BK2451D/BK2452M

Datasheet

BK51 MCU+RF

V1.3

Beken Corporation
3A,1278 Keyuan Rd,Zhangjiang High-Tech Park
Pudong new Distrinct,Shanghai,201203,china
Tel： (86)21 51086811
Fax： (86)21 60871277

Revision History

Version	Date	Author(s)	Description
0.1	Mar. 8,2014	Lizhen	Initial flash version
1.0	May 30, 2014	Lizhen	Define the package for MP
1.1	Oct 22, 2014	Lizhen	Add QFN-32 package
1.2	Oct 24, 2014	Lizhen	Add RF part parameter

Table of Contents

1 Introduction 8
2 Feature 8
2.1 BK2451 8
2.2 BK2452 8
3 Block Diagram 9
BK2451 9
BK2452 9
4 PIN information 10
4.1 BK5100 RAM version (only used for development) 10
4.2 BK2451D 12
4.3 BK2452M-SOP 13
4.4 BK2452M-QFN 14
5 BK51 Micro-Controller 15
5.1 Instruction Set 15
5.2 Memory organization 21
5.2.1 Program memory 21
5.2.2 Data memory 21
5.2.3 General register 23
5.2.4 Bit address space 23
5.2.5 Stack 23
5.2.6 SFR 24
5.3 Power management 27
5.3.1 Work State 28
5.3.2 Wake Up 29
5.4 Interrupt 30
5.4.1 Interrupt priority 30
5.4.2 Interrupt sampling 31
5.4.3 Clear interrupt 32
5.4.4 Register relevant to interrupt 32
6 Reset system 37
7 Clock system 38
8 General I/O 39
8.1 Basic function 39
8.2 Wake up function 39
8.3 Second function of port 40
8.3.1 Sencond function 40
8.3.2 SECOND FUCNTION FOR GPIO 41
9 Watch Dog 42
10 Timer 43
10.1 Timer0 and Time1 43
10.1.1 Mode0 44
10.1.2 Mode1 45
10.1.3 Mode2 45

10.1.4 Mode3 46
10.2 Timer – Rate Conrol 47
10.3 Timer2 47
10.3.1 16-Bit Timer/Counter Mode 49
10.3.2 16-Bit Timer/Counter Mode with Auto-Reload 49
10.3.3 Baud Rate Generator Mode 50
11 UART 51
11.1 Mode0 51
11.2 Mode1 52
11.2.1 Mode1 Baud Rate 52
11.2.2 Mode1 tramsmit 54
11.2.3 Mode1 Receive 54
11.3 Mode2 55
11.4 Mode3 56
12 LBD 57
13 Random Number Generator 58
14 USB2.0 59
14.1 Clock 59
14.2 USB Register Access 59
14.3 ENDPOINT Configuration 59
14.4 Interrupt 61
14.5 FIFO 63
14.5.1 FIFO SFR register 64
14.5.2 FIFO Access 65
14.5.3 FIFO Operation 65
14.6 Device Address 66
14.7 Frame number register 66
14.8 USB power management 66
14.9 Endpoint Buffer 67
15 RF part 68
15.1 Electrical Specifications 69
16 Typical Application Schematic 71
17 Package Information 72
18 Solder Reflow Profile 76
19 Order Information 77
20 Solder Reflow Profile 77
21 Contact Information 78

List of Figures

FIGURE 1BK2451 BLOCK DIAGRAM 9
FIGURE 2BK2452 BLOCK DIAGRAM 9
FIGURE 3 BK510M 10
FIGURE 4 BK2451D 12
FIGURE 5 BK2452M-1 13
FIGURE 6 BK2452M-2 14
FIGURE 7 BK-51 ARCHITECTURE 16
FIGURE 8 MEMORY SPACES 21
FIGURE 9 INTERNAL MEMORY SPACES 22
FIGURE 10 WAKE UP PROCESS 29
FIGURE 11 CLOCK SYSTEM 38
FIGURE 12 TIMER 0/1 – MODES 0 AND 1 45
FIGURE 13 TIMER 0/1 – MODE 2 46
FIGURE 14 TIMER 0 – MODE 3 46
FIGURE 15 TIMER 2 – TIMER/COUNTER WITH CAPTURE 49
FIGURE 16 TIMER 2 – TIMER/COUNTER WITH AUTO-RELOAD 50
FIGURE 17 TIMER 2 – BAUD RATE GENERATOR MODE 50
FIGURE 18 UART MODE0 TRAMSMIT TIMING 52
FIGURE 19 UART MODE0 RECEIVE TIMING 52
FIGURE 20 UART MODE1 TRANSMIT TIMING 54
FIGURE 21 UART MODE1RECEIVE TIMING 54
FIGURE 22 UART MODE2 TRANSMIT TIMING 55
FIGURE 23 UART MODE2 RECEIVE TIMING 56
FIGURE 4 CLASSIFICATION REFLOW PROFILE 76

List of tables

TABLE 1 PIN DEFINITION 11
TABLE 2 PIN DEFINITION 13
TABLE 3 PIN DEFINITION 13
TABLE 4 PIN DEFINITION 15
TABLE 5 SFR REGISTER 27
TABLE 6 POWER MANAGEMENT REGISTER 27
TABLE 7 INTERRUPT PRIORITY 31
TABLE 8 INTERRUPT TRIGGER MODE 31
TABLE 9 IE REGISTER 33
TABLE 10 IP REGISTER 33
TABLE 11 EXIF REGISTER 34
TABLE 12 EICONREGISTER 34
TABLE 13 EIE REGISTER 35
TABLE 14 EIP REGISTER 36
TABLE 15 PORT0 SECOND FUNCTION 40
TABLE 16 PORT1 SECOND FUNCTION 40
TABLE 17 PORT2 SECOND FUNCTION 40
TABLE 18 GPIO FOR 32-PIN PACKAGE 41
TABLE 19 WATCH DOG REGISTER 42
TABLE 20 THE PRESCALE OF WATCH DOG CLOCK 42
TABLE 21 TIMER TMOD REGISTER 44
TABLE 22 TIMER TCON REGISTER 44
TABLE 23TIMER2 T2CON REGISTER 48
TABLE 24 TIMER 2 MODE CONTROL SUMMARY 48
TABLE 25 UART SCON REGISTER 51
TABLE 26 LBD REGISTER 1 57
TABLE 27 LBD REGISTER 2 57
TABLE 28 LBD REGISTER 3 57
TABLE 29 LBD VOTAGE TABLE 57
TABLE 30 USB MSB ENDPOINT ADDRESS 60
TABLE 31 CONFIGURE REGISTER 1 OF ENDPOINT 0 60
TABLE 32 ENDPOINT N CONFIGURE REGISTER 60
TABLE 33 ENDPOINT NCONFIGURE REGISTER 0 61
TABLE 34 USBINT0 INTERRUPT REGISTER 61
TABLE 35 EP_STATUS REGISTER 62
TABLE 36 EP_STATUS REGISTER 62
TABLE 37 USBINT1 INTERRUPT REGISTER 63
TABLE 38 USB_EN0 INTERRUPT ENABLE REGISTER 63
TABLE 39 USB_EN1INTERRUPT ENABLE REGISTER 63
TABLE 40 FIFO EP_RDY REGISTER 64
TABLE 41 FIFO LOWER 8 BITS COUNTER REGISTER 64
TABLE 42 FIFO UPPER 2 BITS COUNTER REGISTER 64
TABLE 43 FIFO EP_HALT REGISTER 65
TABLE 44 DEVICE ADDRESS REGISTER 66
TABLE 45 FRAM_NO_0 LOWER 8 BITS REGISTER 66
TABLE 46 FRAM_NO_0 UPPER 3 BITS REGISTER 66
TABLE 47 USB POWER CONTROL REGISTER 67
TABLE 48 MCU ELECTRICAL SPECIFICATION 70
TABLE 14 SOLDER REFLOW PROFILE 76

1 Introduction

The BK2450 is serial chipset embedded BK51 processor include BK2451 used for dongle and BK2452used for mouse.

2 Feature

2.1 BK2451

- 1.9 V to 3.6 V power supply
- USB2.0 full speed controller(12MHz)
- Bk51 MCU compatible with 8051
- One instruction one period, and 70% instruction no more than 2 period
- 8k bytes OTP for program
- 256 Bytes IRAM and 1k Bytes SRAM
- Embedded three Timer/Counter
- Support UART
- low power consumption, embedded with 32k RC oscillator
- Total 3 GPIO available
- Integrated 2.4G RF transceiver

2.2 BK2452

- 1.9 V to 3.6 V power supply
- Bk51 MCU compatible with 8051
- One instruction one period, and 70% instruction no more than 2 period
- 8k bytes ROM for program
- 256 Bytes IRAM and 1k Bytes SRAM
- Embedded three Timer/Counter
- Random number generator (True)
- Support UART
- low power consumption, embedded with 32k RC oscillator
- Low power detect circuit （LBD）
- Total 18 GPIO available
- Integrated 2.4G RF transceiver

3 Block Diagram

BK2451

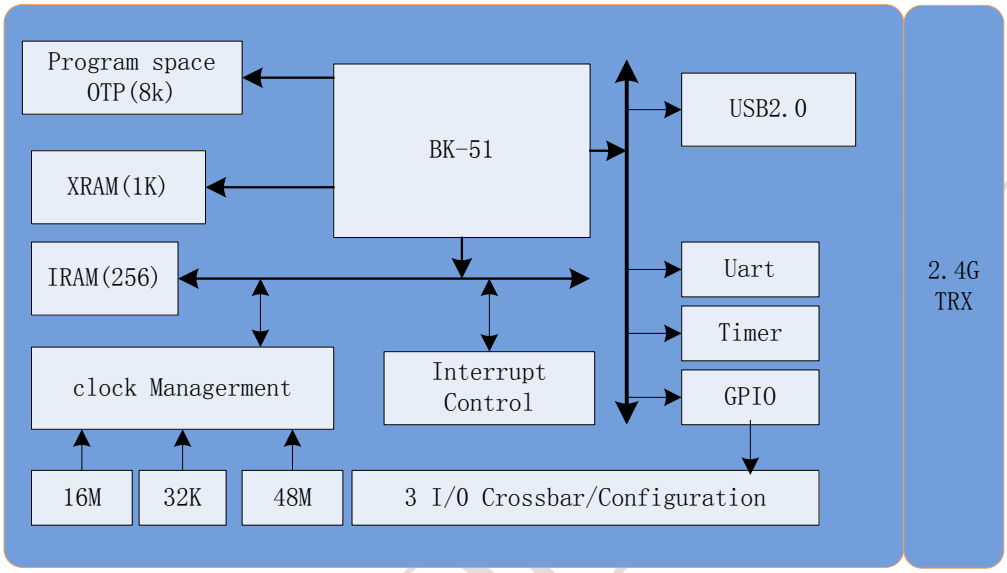


Figure 1BK2451 Block Diagram

BK2452

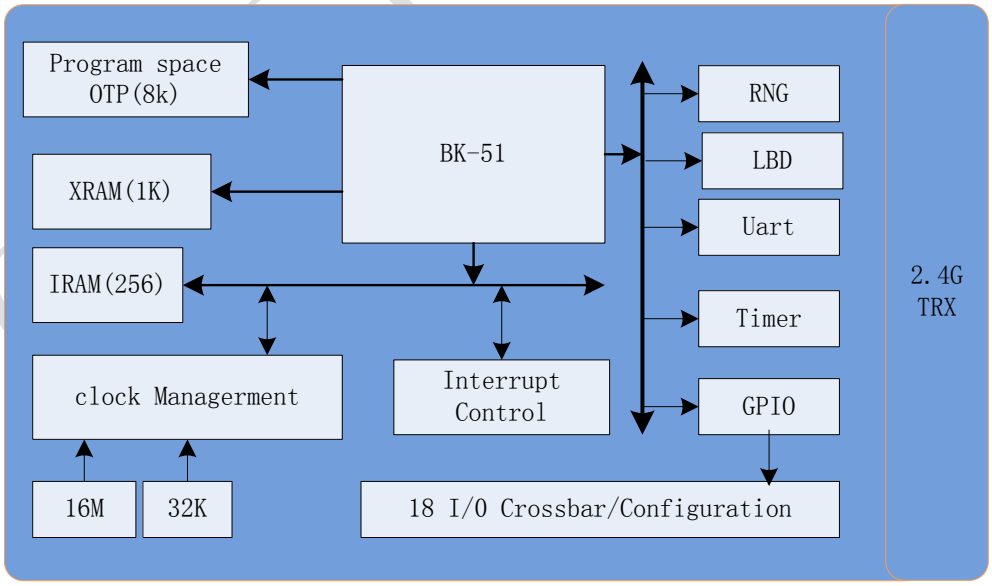


Figure 2BK2452 Block Diagram

4 PIN information

4.1 BK5100 RAM version (only used for development)

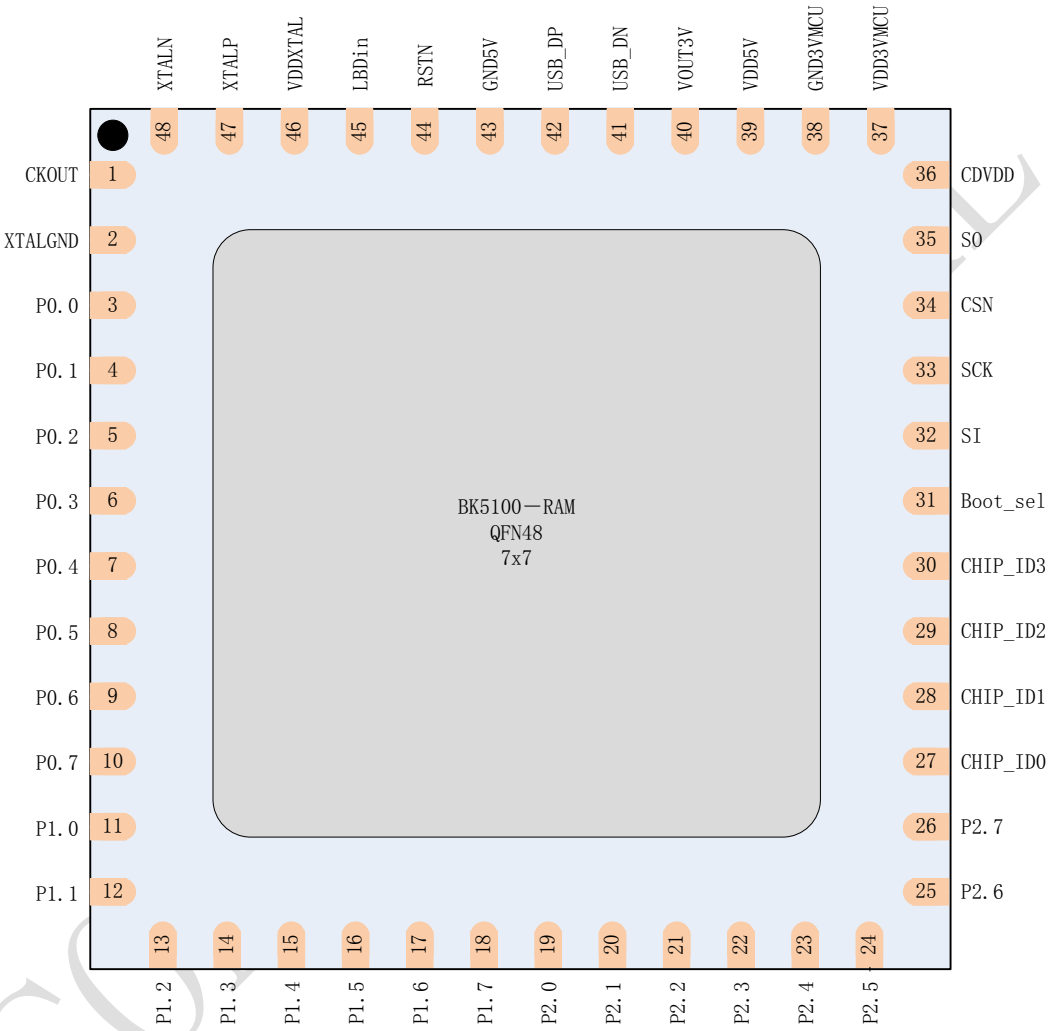


Figure 3 BK510M

Num.	Name	Pin Function	Description
1	CKOUT	CLOCK buffer out	Xtal Clock out used for other chip
2	XTALGND	ground	
3	P0.0	Digital I/O	General I/O，
4	P0.1	Digital I/O	General I/O，
5	P0.2	Digital I/O	General I/O，

6	P0.3	Digital I/O	General I/O，
7	P0.4	Digital I/O	General I/O，
8	P0.5	Digital I/O	General I/O，
9	P0.6	Digital I/O	General I/O，
10	P0.7	Digital I/O	General I/O，
11	P1.0	Digital I/O	General I/O，
12	P1.1	Digital I/O	General I/O，
13	P1.2	Digital I/O	General I/O，
14	P1.3	Digital I/O	General I/O，
15	P1.4	Digital I/O	General I/O，
16	P1.5	Digital I/O	General I/O，
17	P1.6	Digital I/O	General I/O，
18	P1.7	Digital I/O	General I/O，
19	P2.0	Digital I/O	General I/O， or input for UART
20	P2.1	Digital I/O	General I/O， or output for UART
21	P2.2	Digital I/O	General I/O，
22	P2.3	Digital I/O	General I/O，
23	P2.4	Digital I/O	General I/O，
24	P2.5	Digital I/O	General I/O，
25	P2.6	Digital I/O	General I/O，
26	P2.7	Digital I/O	General I/O，
27	CHIP_ID0	Digital input	General input
28	CHIP_ID1	Digital input	General input
29	CHIP_ID2	Digital input	General input
30	CHIP_ID3	Digital input	General input
31	Boot_sel	Digital input	Whether boot program from flash
32	SI	Digital input	Output to external FLASH
33	SCK	Digital input	Output to external FLASH
34	CSN	Digital input	Output to external FLASH
35	SO	Digital input	Input from external FLASH
36	CDVDD	Analog output	power output, connected with decoupling CAP
37	VDD3V	Power supply	3v supply
38	GND	ground	
39	VDD5V	Power	5V supply for USB
40	VOUT3V	Analog output	3v power output, connected with decoupling CAP
41	USB_DN	Digital I/O	USB input N
42	USB_DP	Digital I/O	USB input P
43	GND5V	ground	
44	RSTN	RESET	
45	LBDin	Analog input	Low power detect
46	VDDXTAL	Power supply	3v supply
47	XTALP	Analog output	Oscillator output
48	XTALN	Analog input	Oscillator input

Table 1 PIN definition

4.2 BK2451D

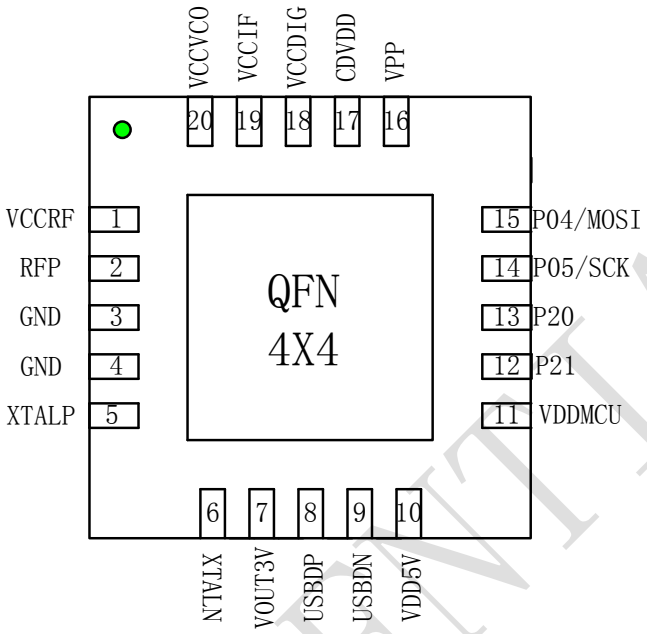


Figure 4 BK2451D

NO.	Name	Pin Function	Description
1	VCCRF	Power supply	3v supply
2	RFP	Antenna input	
3	GND	ground	
4	GND	ground	
5	XTALP	Analog output	Oscillator output
6	XTALN	Analog input	Oscillator input
7	VOUT3V	Analog output	3v power output, connected with decoupling CAP
8	USB_DP	Digital I/O	USB input P
9	USB_DN	Digital I/O	USB input N
10	VDD5V	Power	5V supply for USB
11	VDDMCU	Power supply	3v supply
12	P2.1	Digital I/O	General I/O，
13	P2.0	Digital I/O	General I/O，
14	P05	To BK2425 SCK(fixed)	This pin can be shared for other function depend on software
15	P04	To BK2425 MOSI (fixed)	This pin can be shared for other function depend on software
16	VPP	Power supply	Program mode and 6.5V power supply
17	CDVDD	Analog output	power output, connected with decoupling CAP
18	VCCDIG	Power supply	3v supply for RF part digital

19	VCCIF	Power supply	3v supply
20	VCCVCO	Power supply	3v supply

Table 2 PIN definition

4.3 BK2452M-SOP

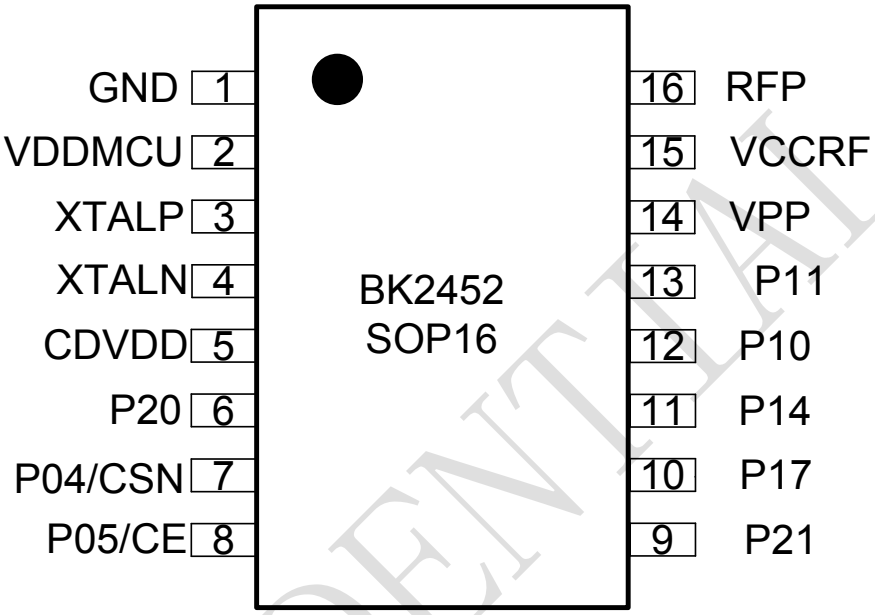


Figure 5 BK2452M-1

NO.	Name	Pin Function	Description
1	GND	ground	
2	VDDMCU	Power supply	power supply
3	XTALP	Analog output	Oscillator output
4	XTALN	Analog input	Oscillator input
5	CDVDD	Analog output	connected with decoupling CAP
6	P20	Digital I/O	General I/O，
7	P04	To BK2425 CSN(fixed)	This pin can be shared for other function depend on software
8	P05	To BK2425 CE(fixed)	This pin can be shared for other function depend on software
9	P2.1	Digital I/O	General I/O，
10	P1.7	Digital I/O	General I/O，
11	P1.4	Digital I/O	General I/O，
12	P1.0	Digital I/O	General I/O，
13	P1.1	Digital I/O	General I/O，
14	VPP	Power supply	Program mode and 6.5V power supply
15	VCCRF	Power supply	Power supply
16	RFP	Antenna input	

Table 3 PIN definition

4.4 BK2452M-QFN

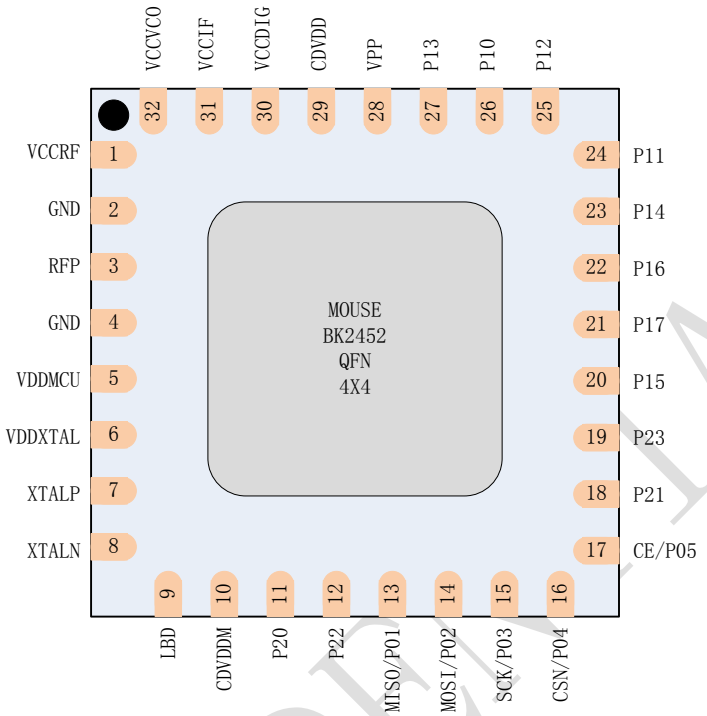


Figure 6 BK2452M-2

NO.	Name	Pin Function	Description
1	VCCRF	Power supply	3v supply
2	GND	ground	
3	RFP	Antenna input	
4	GND	ground	
5	VDDMCU	Power supply	power supply
6	VDDXTAL	Power supply	power supply
7	XTALP	Analog output	Oscillator output
8	XTALN	Analog input	Oscillator input
9	LBDin	Analog input	Low power detect
10	CDVDDMCU	Analog output	connected with decoupling CAP
11	P20	Digital I/O	General I/O，
12	P22	Digital I/O	General I/O，
13	P01	To BK2425 MISO (fixed)	This pin can be shared for other function depend on software
14	P02	To BK2425 MOSI (fixed)	This pin can be shared for other function depend on software
15	P03	To BK2425 SCK(fixed)	This pin can be shared for other function depend on software
16	P04	To BK2425 CSN(fixed)	This pin can be shared for other function depend on software
17	P05	To BK2425 CE(fixed)	This pin can be shared for other function depend on software

18	P2.1	Digital I/O	General I/O，
19	P2.3	Digital I/O	General I/O，
20	P1.5	Digital I/O	General I/O，
21	P1.7	Digital I/O	General I/O，
22	P1.6	Digital I/O	General I/O，
23	P1.4	Digital I/O	General I/O，
24	P1.1	Digital I/O	General I/O，
25	P1.2	Digital I/O	General I/O，
26	P1.0	Digital I/O	General I/O，
27	P1.3	Digital I/O	General I/O，
28	VPP	Power supply	Program mode and 6.5V power supply
29	CDVDD	Analog output	power output, connected with decoupling CAP
30	VCCDIG	Power supply	Power supply for RF part digital
31	VCCIF	Power supply	Power supply
32	VCCVCO	Power supply	Power supply

Table 4 PIN definition

5 BK51 Micro-Controller

5.1 Instruction Set

The core of MCU is BK-51 processor。All BK-51 instructions are binary code compatible and perform the same functions that they do in the industry standard 8051. The effects of these instructions on bits, flags, and other status functions are identical to the industry-standard 8051. However, the timing of the instructions is different, both in terms of number of clock cycles per instruction cycle and timing within the instruction cycle.

BK-51 has comprised all the peripheral equipment of the standard 8051.It include three 16bits timer, one duplex UART, one enhanced SPI, 256 bytes internal RAM, 128 bytes SFR and 32 general I/O port. But it does not include JTAG.

Besides these standard architecture and peripheral equipment, BK-51 has other custom-build peripheral equipment and function, which increase its ability greatly.

BK-51 feature：

- Compatible with MCS-51
- 16MHz frequency， can work at 16M/8M/4M/2M Hz frequency
- 80% instruction consume 1~3 clock period only
- 256 bytes internal RAM
- expanded interrupt deal system
- Pwer managerment mode
- Security for program and storage

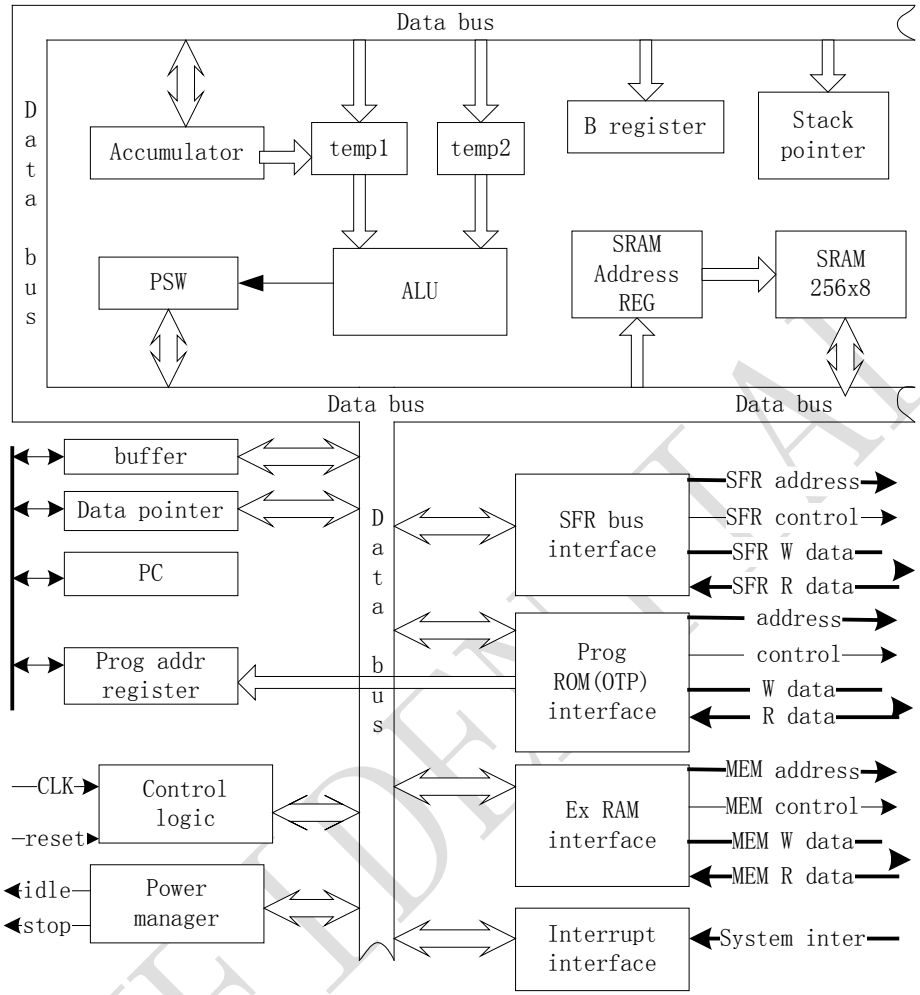


Figure 7 BK-51 architecture

Mnemonic	description	Byte	clock period
Arithmetic			
ADD A,Rn	Add register to A	1	1
ADD A,direct	Add direct byte to A	2	2
ADD A,@Ri	Add data memory to A	1	3
ADD A,#data	Add immediate to A	2	2
ADDC A,Rn	Add register to A with carry	1	1
ADDC A,direct	Add direct byte to A with carry	2	2
ADDC A,@Ri	Add data memory to A with carry	1	2
ADDC A,#data	Add immediate to A with	2	2

	carry		
SUBB A,Rn	Subtract register from A with borrow	1	1
SUBB A,direct	Subtract direct byte from A with borrow	2	2
SUBB A,@Ri	Subtract data memory from A with borrow	1	3
SUBB A,#data	Subtract immediate A with borrow	2	2
INC A	Increment A	1	1
INC Rn	Increment register	1	2
INC direct	Increment direct byte	2	3
INC @Ri	Increment data memory	1	3
DEC A	Decrement A	1	1
DEC Rn	Decrement register	1	2
DEC direct	Increment direct byte	2	3
DEC @Ri	Decrement data memory	1	3
INC DPTR	Increment data pointer	1	1
MUL AB	Multiply A by B	1	2
DIV AB	Divide A by B	1	8
DA A	Decimal adjust A	1	1
Logical			
ANL A,Rn	AND register to A	1	1
ANL A,direct	AND direct byte to A	2	2
ANL A,@Ri	AND data memory to A	1	2
ANL A,#data	AND immediate to A	2	2
ANL direct,A	AND A to direct byte	2	3
ANL direct,#data	AND immediate data to direct byte	3	4
ORL A,Rn	OR register to A	1	1
ORL A,direct	OR direct byte to A	2	2
ORL A,@Ri	OR data memory to A	1	2
ORL A,#data	OR immediate to A	2	2
ORL direct,A	OR A to direct byte	2	3
ORL direct,#data	OR immediate data to direct byte	3	4
XRL A,Rn	Exclusive-OR register to A	1	1
XRL A,direct	Exclusive-OR direct byte to A	2	2
XRL A,@Ri	Exclusive-OR data memory to A	1	2
XRL A,#data	Exclusive-OR immediate to A	2	2
XRL direct,A	Exclusive-OR A to direct	2	3

	byte		
XRL direct,#data	Exclusive-OR immediate to direct byte	3	4
CLR A	Clear A	1	1
CPL A	Complement A	1	1
RL A	Rotate A left	1	1
RLC A	Rotate A left through carry	1	1
RR A	Rotate A right	1	1
RRC A	Rotate A right through carry	1	1
SWAP A	Swap nibbles of A	1	1
Data transfer			
MOV A,Rn	Move register to A	1	1
MOV A,direct	Move direct byte to A	2	2
MOV A,@Ri	Move data memory to A	1	3
MOV A,#data	Move immediate to A	2	2
MOV Rn,A	Move A to register	1	2
MOV Rn,direct	Move direct byte to register	2	3
MOV Rn,#data	Move immediate to register	2	3
MOV direct,A	Move A to direct byte	2	3
MOV direct,Rn	Move register to direct byte	2	3
MOV direct,direct	Move direct byte to direct byte	3	4
MOV direct,@Ri	Move data memory to direct byte	2	3
MOV direct,#data	Move immediate to direct byte	3	4
MOV @Ri,A	MOV A to data memory	1	2
MOV @Ri,direct	Move direct byte to data memory	2	4
MOV @Ri,#data	Move immediate to data memory	2	4
MOV DPTR,#data16	Move 16 bits constant into DPTR	3	3
MOVC A,@A+DPTR	Move code byte relative DPTR to A	1	4
MOVC A,@A+PC	Move code byte relative PC to A	1	4
MOVX A,@Ri	Move external data (A8) to A	1	2
MOVX @Ri,A	Move A to external data (A8)	1	3
MOVX A,@DPTR	Move external data (A16) to A	1	1
MOVX @DPTR,A	Move A to external data (A16)	1	1
PUSH direct	Push direct byte onto stack	2	5

POP direct	Pop direct byte from stack	2	5
XCH A,Rn	Exchange A and register	1	2
XCH A,direct	Exchange A and direct byte	2	3
XCH A,@Ri	Exchange A and data memory	1	3
XCHD A,@Ri	Exchange A and data memory nibble	1	3
Boolean			
CLR C	Clear carry	1	1
CLR bit	Clear direct bit	2	3
SETB C	Set carry	1	1
SETB bit	Set direct bit	2	3
CPL C	Complement carry	1	1
CPL bit	Complement direct bit	2	3
ANL C,bit	AND direct bit to carry	2	2
ANL C,/bit	AND direct bit inverse to carry	2	2
ORL C,bit	OR direct bit to carry	2	2
ORL C,/bit	OR direct bit inverse to carry	2	2
MOV C,bit	Move direct bit to carry	2	2
MOV bit,C	Move carry to direct bit	2	3
JC rel	Jump on carry =1	2	3/4
JNC rel	Jump on carry =0	2	3/4
JB bit,rel	Jump on direct bit= 1	3	4/5
JNB bit,rel	Jump on direct bit= 0	3	4/5
JBC bit,rel	Jump on direct bit= 1 and clear	3	4/5
Branching			
ACALL addr11	Absolute call to subroutine	2	3
LCALL addr16	Long call to subroutine	3	4
RET	Return from subroutine	1	6
RETI	Return from interrupt	1	6
AJMP addr11	Absolute jump unconditional	2	3
LJMP addr16	Long jump unconditional	3	4
SJMP rel	Short jump (relative address)	2	4
JMP @A+DPTR	Jump indirect relative DPTR	1	4
JZ rel	Jump on accumulator= 0	2	3/4
JNZ rel	Jump on accumulator /= 0	2	3/4
CJNE A,direct,rel	Compare A, direct JNE relative	3	3/4
CJNE A,#data,rel	Compare A, immediate JNE relative	3	3/4
CJNE Rn,#data,rel	Compare reg, immediate JNE relative	3	3/4

CJNE @Ri,#data,rel	Compare ind, immediate JNE relative	3	5/6
DJNZ Rn,rel	Decrement register, JNZ relative	2	3/4
DJNZ direct,rel	Decrement direct byte, JNZ relative	3	4/5
NOP	No operation	1	1

80% instruction cycles in the BK-51 are 1~3 clock cycles in length, as opposed to the12 clock cycles per instruction cycle in the standard 8051. This translates to a more than 4X improvement in execution time for most instructions.

The architecture eliminates redundant bus states and implements parallel execution of fetch and execution phases. Most of the one-byte instructions are performed in one single cycle. The MCU uses 1 clock per cycle.

BK-51 can executes the normal 8051 instruction set, so you can use the standard 8051 tools to develop your application. All the instructions are equal to 8051's binary codes and function, includes operate code, addressing mode, affect on PSW. But the timing of instruction is different with 8051.

In many 8051 products, the machine period is different with clock period, one machine period last on 2~12 clock period, but BK-51 only has clock period, all the instructions are based on the clock period.

5.2 Memory organization

The memory organization of BK-51 is similar with 8051. It has two independent spaces: data memory and program memory. You can access them with different instructions.

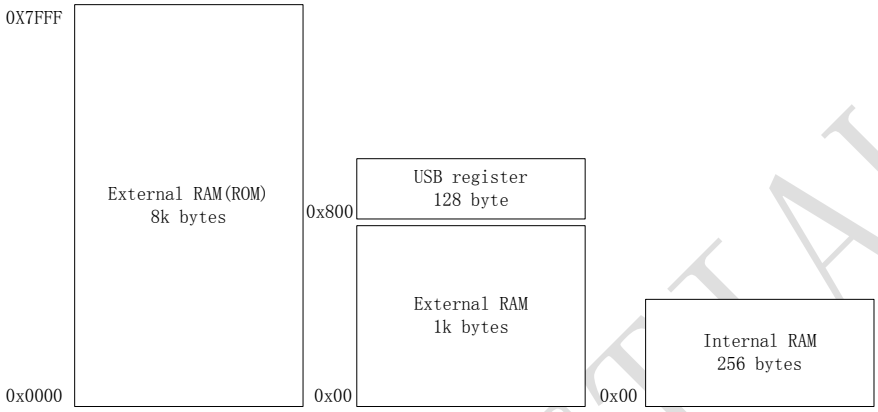


Figure 8 Memory Spaces

5.2.1 Program memory

8k bytes ROM (one time program) memory is used for storing program.

Also, RAM version will be provided to customer for development. You can download program into external flash firstly, then BK5100 will read the program into RAM from flash when power up.

5.2.2 Data memory

There are 2k bytes external memories and 256 bytes internal memory in BK-51 as showed in Figure 8.

The internal RAM (Figure 9) consists of 128 bytes of registers and scratchpad memory accessible through direct or indirect addressing (iram_addr addresses 00h–7Fh) Upper 128 bytes of scratchpad memory accessible through indirect addressing (iram_addr addresses 80h–FFh). 128 special function registers (SFRs) accessible through direct addressing (sfr_addr addresses 80h–FFh)

The SFRs and the upper 128bytes of RAM share the same address range (80h-FFh). However, the actual address space is separate and is differentiated by the type of addressing. Direct addressing accesses the SFRs on the sfr_bus, indirect addressing accesses the upper 128bytes of RAM on the iram_bus.

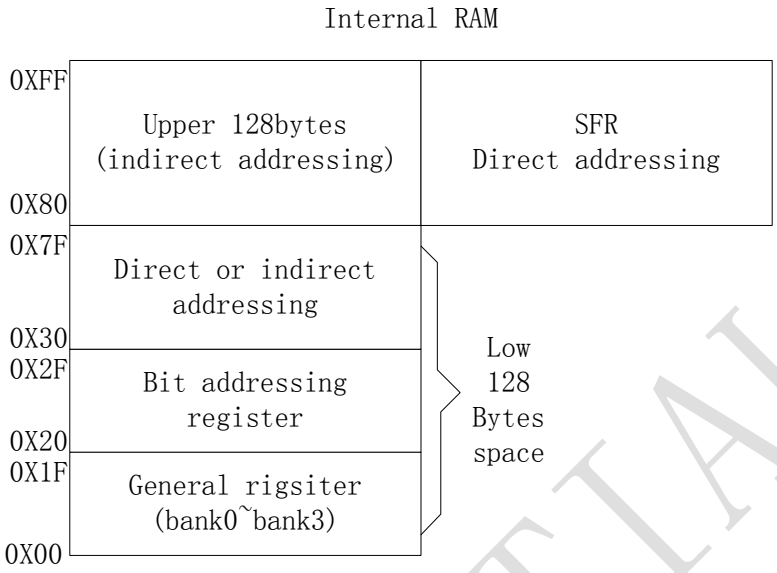


Figure 9 Internal memory spaces

2k bytes external ram are embedded in BK5100, they are mapped to the XRAM address from 0x0000 to 0x07FF. Also, any part of this XRAM can be configured as the buffer for USB endpoints. The configured USB space can be accessed by USB module or MCU part, and the hardware arbiter deal the conflict between them without customer operation.

5.2.3 General register

The lower 128 bytes are organized as shown in Figure 4. The lower 32bytes form four banks of eight registers (R0–R7). Two bits on the program status word (RS0（PSW.3）and RS1（PSW.4）) select which bank is in use. This allow fast switch when entering subroutine or interrupt program. R0 and R1 are used for indirect addressing register.

5.2.4 Bit address space

The next 16bytes form a block of bit-addressable memory space at bit addresses 00h–7Fh. All of the bytes in the lower 128 bytes are accessible through direct or indirect addressing on the iambus. Every bit has a bit address from 0x00 to 0x7F. For example, the bit 0 of address 0x20 has a bit address 0x00, the bit 7 of address 0x20 has a bit address 0x07, the bit 7 of address 0x2F has a bit address 0x7F. The type of instruction decides the addressing mode (bit addressing or byte addressing).

MCS-51™ use XX.B to represent bit address, XX means byte address，and B means bit address. For example:

MOV C， 23h . 0

Move the Boolean variable in 0x23 to carry flag. (The bit 0 of 0x23h)

5.2.5 Stack

The stack for program can locate any position of the 256 bytes memory. Stack pointer is used to point the stack area, SP point to the last position. The next stack data is placed at SP+1, and then SP increased by 1. The initial stack pointer address is 0x07, so the first stack data is placed at 0x08, which is the first register R0. The max depth of the stack can be 256 bytes.

5.2.6 SFR

REGISTER	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0	addr	reset value
P0	-	-	-	-	-	-	-	-	0x80	0xff
SP	-	-	-	-	-	-	-	-	0x81	
DPL0	-	-	-	-	-	-	-	-	0x82	
DPH0	-	-	-	-	-	-	-	-	0x83	
DPL1	-	-	-	-	-	-	-	-	0x84	
DPH1	-	-	-	-	-	-	-	-	0x85	
DPS								SEL	0x86	
PCON	SMOD	EUSB	CMD_RST	Latch_en	deep_sleep	OSC32k	RC32k	IDLE	0x87	0x02
TCON	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	0x88	
TMOD	GATE	C/T	M1	M0	GATE	C/T	M1	M0	0x89	
TL0	-	-	-	-	-	-	-	-	0x8A	
TL1	-	-	-	-	-	-	-	-	0x8B	
TH0	-	-	-	-	-	-	-	-	0x8C	
TH1	-	-	-	-	-	-	-	-	0x8D	
CKCON	CKDIV1	CKDIV0	T2M	T1M	T0M	/	/	/	0x8E	0
CLK_EN_CFG		timer_en	uart_en					wdt_en	0x8F	0
P1	-	-	-	-	-	-	-	-	0x90	0xFF
EXIF	IE9	IE8	IE7	IE6	IE5	IE4	IE3	IE2	0x91	0x08
MPAGE	-	-	-	-	-	-	-	-	0x92	
EICON	/	/	EPFI	PFI	/	/	/	/	0x93	0
									0x94	0
									0x95	0
									0x96	0
WDT	/	/				PS2	PS1	PS0	0x97	0
SCON0	SM0	SM1	SM2	REN	TB8	RB8	TI	RI	0x98	0
SBUF0	-	-	-	-	-	-	-	-	0x99	0
P0_PU									0x9A	0
P1_PU									0x9B	0
P2_PU									0x9C	0
									0x9D	0
									0x9E	0
P2	-	-	-	-	-	-	-	-	0xA0	0xFF
										0
									0xA2	0
									0xA3	0
									0xA4	0
									0xA5	0
									0xA6	0

									0xA7	0
IE	EA		ET2	ES0	ET1	EX1	ET0	EX0	0xA8	0
IP		PPFI	PT2	PS0	PT1	PX1	PT0	PX0	0xA9	0x80
P0_IOSEL	-	-	-	-	-	-	-	-	0xAA	0
P1_IOSEL	-	-	-	-	-	-	-	-	0xAB	0
P2_IOSEL	-	-	-	-	-	-	-	-	0xAC	0
		-	-	-	-	-	-	-	0xAD	0
									0xAE	0
	-	-	-	-	-	-	-	-	0xB0	0xFF
P0_PD	-	-	-	-	-	-	-	-	0xB1	0
P1_PD	-	-	-	-	-	-	-	-	0xB2	0
P2_PD	-	-	-	-	-	-	-	-	0xB3	0
	-	-	-	-	-	-	-	-	0xB4	0
	-	-	-	-	-	-	-	-	0xB5	0
									0xB6	0
									0xB7	0
									0xB8	0
									0XB9	0xEC
P0_OPDR	-	-	-	-	-	-	-	-	0xBA	0xFF
P1_OPDR	-	-	-	-	-	-	-	-	0xBB	0xFF
P2_OPDR	-	-	-	-	-	-	-	-	0xBC	0xFF
		-	-	-	-	-	-	-	0xBD	0xFF
		-	-	-	-	-	-	-	0xBE	0xFF
	-	-	-	-	-	-	-	-	0xC0	0xFF
									0xC1	0
									0xC2	0
									0xC3	0
P0_WUEN									0xC4	0
P1_WUEN									0xC5	0
P2_WUEN									0xC6	0
									0xC7	0
T2CON	TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/T2	CP/RL2	0xC8	0
									0xC9	0
RCAP2L	-	-	-	-	-	-	-	-	0xCA	0
RCAP2H	-	-	-	-	-	-	-	-	0xCB	0
TL2	-	-	-	-	-	-	-	-	0xCC	0
TH2	-	-	-	-	-	-	-	-	0xCD	0
									0xCE	0
									0xCF	0
PSW	CY	AC	F0	RS1	RS0	OV	F1	P	0xD0	0
									0xD1	0
									0xD2	0
									0xD3	0
									0xD4	0
									0xD5	0

									0xD6	0
									0xD7	0
ANA_REG									0xD8	
ANA_REG									0xD9	
ANA_REG									0xDa	
ANA_REG									0xDb	
ANA_REG									0xDc	
ANA_REG									0xDd	
ANA_REG									0xDe	
ANA_REG									0xDf	
ACC	-	-	-	-	-	-	-	-	0xE0	0
P0_WUMOD									0xE1	
P1_WUMOD									0xE2	
P2_WUMOD									0xE3	
									0xE4	
									0XE5	
									0XE6	
PALT	S_S0	S_T0	S_T1	S_T2	/	/	/	/	0xE7	0
EIE	EX9	EX8	EX7	EX6	EX5	EX4	EX3	EX2	0xE8	0xF0
EIP	PX9	PX8	PX7	PX6	PX5	PX4	PX3	PX2	0xE9	0xF0
EXINT_MOD	IT9	IT8	IT7	IT6	IT5	IT4	IT3	IT2	0xEA	0xFF
RNG_DAT									0xEB	
RNG_CTL	RNG_PWD		RNG_RDY						0xEC	
									0xED	
									0xEE	
									0xEF	
	-	-	-	-	-	-	-	-	0xF0	0
	-	-	-	-	-	-	-	-	0xF1	0
	-	-	-	-	-	-	-	-	0xF2	0
	-	-	-	-	-	-	-	-	0xF3	0
	-	-	-	-	-	-	-	-	0xF4	0
	-	-	-	-	-	-	-	-	0xF5	0
	-	-	-	-	-	-	-	-	0xF6	0
									0xF7	
									0xF8	
									0xF9	0x01
									0xFA	0
									0xFB	0
CHIP_ID_H										
CHIP_ID_L									0xFF	

Table 5 SFR Register

5.3 Power management

PCON	7	6	5	4	3	2	1	0
0x87	SMOD	EUSB	CMD_RST	Latch_en	Deep_sleep	OSC32K_sel	RC32k_sel	IDLE

Table 6 power management register

SMOD: SMOD0 – Serial Port 0 baud rate doublers enable. When SMOD0=1, the baud rate for Serial Port 0 is doubled.

EUSB: R/W by software only. USB enable, the 48MHz clock will exist when EUSB=1.

CMD_RST: Write 1 to reset MCU (not include RF part).

RC32k_sel: System will select RC32k clock when write 1 to this position. Interrupts and software can clear it.（when reset, this bit will be reset to1and system enter slow clock mode）. In this state, the system clock changed to 32K RC clock, so the power consumption is very low.

OSC32k_sel: System will select OSC32k clock when write 1 to this position. Interrupts and software can clear it. In this state, the system clock changed to 32K OSC clock, so the power consumption will decrease evidently. Pls note that OSC32k clock is more accurate than RC 32k clock, but need more power consumption.

IDLE: When set by software, system enter stop mode, and it can only be wake up by enabled interrupt.(Clear it to 0). In this state, the most of clocks are shut down for power saving.

Note: set RC32k_sel and IDLE bit simultaneously can get the lowest power consumption, and in this state, all the register settings are retained.

CLK_EN_CFG	7	6	5	4	3	2	1	0
0x8F	**adc_en**	**timer_en**	**uart_en**	**pwm_en**	**spi_en**	**i2c_en**	**des_en**	**mdu_en**

CLK_EN_CFG: this register can use to power on or off all the peripheral equipment clocks for saving power.

adc_en : ADC clock enable or not (1 enable) (not avaivlabe for BK5100)

timer_en : TIMER 0 /1 /2 clock enable or not (1 enable)

uart_en : UART clock enable or not (1 enable)

pwm_en : PWM clock enable or not (1 enable) (not avaivlabe for BK5100)

spi_en : SPI clock enable or not (1 enable) (not avaivlabe for BK5100)

i2c_en : I2C clock enable or not (1 enable) (not avaivlabe for BK5100)

des_en : DES clock enable or not (1 enable) (not avaivlabe for BK5100)

mdu_en : MDU clock enable or not (1 enable) (not avaivlabe for BK5100)

5.3.1 Work State

5.3.1.1 IDLE MODE

An instruction that sets the IDLE bit (PCON.0) causes the BK-51 to enter idle mode when that instruction completes. In idle mode, CPU processing is suspended, internal registers maintain their current data, and the idle_mode_n output is activated. However, unlike the standard 8051, the clock is not disabled internally.

Activation of any enabled interrupt causes the hardware to clear the IDLE bit and terminate idle mode.

In idle mode, the power consumption is decreased evidently.

5.3.1.2 SLEEP MODE

An instruction that sets the STOP bit (PCON.1) causes the BK-51 to enter stop mode when that instruction completes. In sleep mode, the system clock change from 16MHz to 32KHz, and the clock 48MHz used by USB stopped. The power consumption decreases to a lower level.

If write 1 to PCON.1 at this time, the system will enter deep sleep mode, and the power consumption of MCU will decreased evidently.

After reset, the system will enter sleep mode immediately.

Note: You should always clear PCON.6 to 0 to save current when USB module doesn't need work at any time.

As showed above, the IDLE bit decide CPU run or not, the SLEEP bit decide the system clock source. (16MHz or 32KHz)

5.3.2 Wake Up

5.3.2.1 Wake Up from sleep moe

S_WUEN	port	7	6	5	4	3	2	1	0
0xc4	0	En/dis							
0xc5	1								
0xc6	2								
0xc7	3								
0xc9	4								

When the MCU entered IDLE/SLEEP mode, all the 40 ports pin can be used to wake up the MCU separately. Configure the corresponding SFR bit can enable or disenable the wake up function.

The process wake up from sleep mode is showed in next figure.

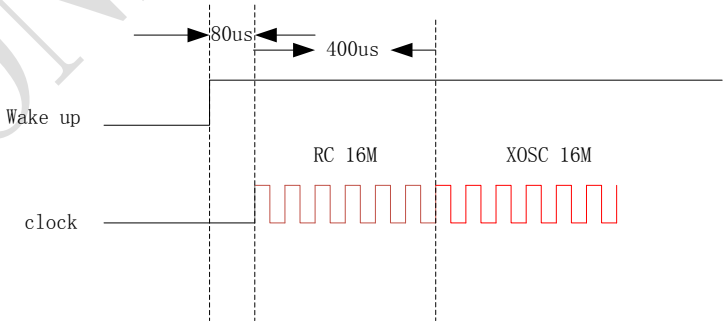


Figure 10 wake up process

After wake up, RC 16M clock will spend 80us to wake up, and after 400us, the clock source will switch to XOSC 16M automatically. RC 16M clock is not very accurate, so, during this period, you can only run ordinary MCU instruction, but can not send or receive RF package.

Pls note that: The RF part will also resume 120us for PLL locking after power up RF part. So, if you want to send/receive package through RF, you should wait 600us after wake up from sleep mode.

5.4 Interrupt

BK-51 has an advanced interrupt controller with 15 sources. Software can set any interrupt source through write 1 to the corresponding position. When an enabled interrupt occurs, the MCU vectors to the address of the interrupt service routine (ISR) associated with that interrupt. The MCU executes the ISR to completion unless another interrupt of higher priority occurs.

The interrupt source, interrupt address, interrupt priority and control bit are showed in next table. Please refer to the relevant chapter to get the detail interrupt condition and interrupt status of external interrupt.

5.4.1 Interrupt priority

Interrupt name	description	Flag bit	Enable bit	Priority control	Interrupt vector	Interrupt number	EXIST
int_0n	External interrupt 0	TCON.1	IE.0	IP.0	03h	0	
TF0	Timer 0 interrupt	TCON.5	IE.1	IP.1	0bh	1	
int1_n	External interrupt 1	TCON.3	IE.2	IP.2	13h	2	
TF1	Timer 1 interrupt	TCON.7	IE.3	IP.3	1bh	3	
TI or RI	Serial port interrupt	SCON.0(RI) SCON.1(TI)	IE.4	IP.4	23h	4	
TF2 or EXF2	Timer 2 interrupt	T2CON.7(TF2) T2CON.6(EXF2)	IE.5	IP.5	2bh	5	
PFI	Votage too low interrupt	EICON[4]	EICON[5]	IP.6	33h	6	
int2	SPI0 interrupt	EXIF[0]	EIE.0	EIP[0]	43h	8	N
int3_n	I2C interrupt	EXIF[1]	EIE.1	EIP[1]	4bh	9	N
int4	USB interrupt	EXIF[2]	EIE.2	EIP[2]	53h	10	
int5_n	RF interrupt	EXIF[3]	EIE.3	EIP[3]	5bh	11	N

int6	Rtc interrupt	EXIF[4]	EIE.4	EIP[4]	63h	12	N
int7_n	PWM interrupt	EXIF[5]	EIE.5	EIP[5]	6bh	13	N
int8	ADC interrupt	EXIF[6]	EIE.6	EIP[6]	73h	14	N
int9_n	AES interrupt	EXIF[7]	EIE.7	EIP[7]	7bh	15	N

Table 7 interrupt priority

You can assign priority level (high or low) to each interrupt as listed in Table 5. Simultaneous interrupts with the same priority level (for example, both high) are resolved according to their arrived time.

Once an interrupt is being serviced, only an interrupt of higher priority level can interrupt the service routine of the interrupt currently being serviced.

5.4.2 Interrupt sampling

The int0_n and int1_n signals are both active low and can be programmed to be either edge-sensitive or level-sensitive, through the IT0 and IT1 bits in the TCON SFR. For example, when IT0= 0, int0_n is level-sensitive and the BK51 sets the IE0 flag when the int0_n pin is sampled low. When IT0 = 1, int0_n is edge-sensitive and the BK51 sets the IE0 flag when the int0_n pin is sampled high then low on consecutive samples.

TF0, TF1 and TF2 are the interrupt used for timer. TI (or RI) is used for UART. PFI used for low battery detector circuit.

The remaining external interrupts can be configured as edge-sensitive interrupt or level-sensitive interrupt; int2, int4, int6 and int8 are active high, int3_n, int5_n, int7_n and int9_n are active low.

Note: External interrupts 2, 3, 4, 5, 6, 7, 8, 9 had been used by BK5100 function module, so customer can't use them again. Customer can configure their trigger mode only.

EXINT_MOD	7	6	5	4	3	2	1	0
0xEA	IT9	IT8	IT7	IT6	IT5	IT4	IT3	IT2

Table 8 interrupt trigger mode

IT9 : when IT9 = 1, int9_n(DES) edge sensitive; when IT9 = 0，level sensitive.
IT8 : when IT8 = 1, int8(ADC) edge sensitive; when IT8 = 0，level sensitive.
IT7 : when IT7 = 1, int7_n(PWM) edge sensitive; when IT7 = 0，level sensitive.
IT6 : when IT6 = 1, int6(RTC) edge sensitive; when IT6 = 0，level sensitive.
IT5 : when IT5 = 1, int5_n(RF) edge sensitive; when IT5 = 0，level sensitive.
IT4 : when IT4 = 1, int4(USB2.0) edge sensitive; when IT4 = 0，level sensitive.
IT3 : when IT3 = 1, int3_n(I2C) edge sensitive; when IT3 = 0，level sensitive.

IT2 : when IT2 = 1, int2(SPI) edge sensitive; when IT2 = 0，level sensitive.

5.4.3 Clear interrupt

When interrupt happens, the interrupt flag of Timer0, timer1, int0 and int1 would be cleared by hardware automatically after the CPU vectors to the address of the interrupt service routine (ISR) associated with that interrupt. The other interrupt must be cleared by customer, or the CPU will respond to the interrupt always.

5.4.4 Register relevant to interrupt

The following SFRs are associated with interrupt control:
IE – SFR A8h
IP – SFR A9h
EXIF – SFR 91h
EICON – SFR 93h
EIE – SFR E8h
EIP – SFR E9h

The IE and IP SFRs provide interrupt enable and priority control for the standard interrupt unit, as with the standard 8051.

The EXIF, EICON, EIE, and EIP registers provide flags, enable control, and priority control for the optional extended interrupt unit.

IE register

bit	definition
IE.7	EA – Global interrupt enable. Controls masking of all interrupts except power-fail interrupt (pfi). EA=0 disables all interrupts (EA overrides individual interrupt enable bits). When EA=1, each interrupt is enabled or masked by its individual enable bit.
IE.6	/
IE.5	ET2 – Enable Timer 2 interrupt. ET2=0 disables Timer 2 interrupt (TF2). ET2=1 enables interrupts generated by the TF2 or EXF2 flag. If Timer 2 is not implemented (timer2=0), ET2 is present but not used.
IE.4	ES0 – Enable Serial Port interrupt. ES0=0 disables Serial Port interrupts (TI and RI). ES0=1 enables interrupts generated by the TI or RI flag.
IE.3	ET1 – Enable Timer1 interrupt. ET1=0 disables Timer1 interrupt (TF1). ET1=1 enables interrupts generated by the TF1 flag.
IE.2	EX1 – Enable external interrupt 1. EX1=0 disables external interrupt 1 (int1_n). EX1=1 enables interrupts generated by the int1_n pin.
IE.1	ET0 – Enable Timer 0 interrupt.

	ET0=0 disables Timer 0 interrupt (TF0). ET0=1 enables interrupts generated by the TF0 flag.
IE.0	EX0 – Enable external interrupt 0. EX0=0 disables external interrupt 0 (int0_n). EX0=1 enables interrupts generated by the int0_n pin.

Table 9 IE register

IP register

bit	definition
IP.7	/
IP.6	PPFI： voltage too low interrupt priority control 0： PFI low priority 1： PFI high priority
IP.5	PT2 – Timer 2 interrupt priority control. PT2= 0 sets Timer 2 interrupt (TF2) to low priority. PT2=1 sets Timer 2 interrupt to high priority. If Timer 2 is not implemented (timer2= 0), PT2 is present but not used.
IP.4	PS0 – Serial Port interrupt priority control. PS0= 0 sets Serial Port interrupt (TI or RI) to low priority. PS0=1 sets Serial Port interrupt to high priority.
IP.3	PT1 – Timer1 interrupt priority control. PT1= 0 sets Timer1 interrupt (TF1) to low priority. PT1=1 sets Timer1 interrupt to high priority.
IP.2	PX1 – External interrupt 1 priority control. PX1= 0 sets external interrupt 1 (int1_n) to low priority. PT1= 1 sets external interrupt 1 to high priority.
IP.1	PT0 – Timer 0 interrupt priority control. PT0= 0 sets Timer 0 interrupt (TF0) to low priority. PT0=1 sets Timer 0 interrupt to high priority.
IP.0	PX0 – External interrupt 0 priority control. PX0= 0 sets external interrupt 0 (int0_n) to low priority. PT0= 1 sets external interrupt 0 to high priority.

Table 10 IP register

EXIF register

bit	definition
EXIF.7	IE9 – External interrupt 9 flag. IE9=1 indicates that an interrupt was detected at the int9_n pin. IE9 must be cleared by software. Setting IE9 in software generates an interrupt, if enabled.
EXIF.6	IE8 – External interrupt 8 flag. IE8=1 indicates that an interrupt was detected at the int8 pin. IE8 must be cleared by software. Setting IE8 in software generates an interrupt, if enabled.

EXIF.5	IE7– External interrupt 7 flag. IE7=1 indicates that a interrupt was detected at the int7_n pin. IE7 must be cleared by software. Setting IE7 in software generates an interrupt, if enabled.
EXIF.4	IE6 – External interrupt 6 flag. IE6=1 indicates that an interrupt was detected at the int6 pin. IE6 must be cleared by software. Setting IE6 in software generates an interrupt, if enabled.
EXIF.3	IE5– External interrupt 5 flag. IE5=1 indicates that an interrupt was detected at the int5_n pin. IE5 must be cleared by software. Setting IE5 in software generates an interrupt, if enabled.
EXIF.2	IE4 – External interrupt 4 flag. IE4=1 indicates that an interrupt was detected at the int4 pin. IE4 must be cleared by software. Setting IE4 in software generates an interrupt, if enabled.
EXIF.1	IE3– External Interrupt 3 flag. IE3=1 indicates that an interrupt was detected at the int3_n pin. IE3 must be cleared by software. Setting IE3 in software generates an interrupt, if enabled.
EXIF.0	IE2 – External interrupt 2 flag. IE2=1 indicates that an interrupt was detected at the int2 pin. IE2 must be cleared by software. Setting IE2 in software generates an interrupt, if enabled.

Table 11 EXIF register

EICON （0x93）

bit	definition
EICON.7	/
EICON.6	/
EICON.5	EPFI – Enable power-fail interrupt. EPFI= 0 disables power-fail interrupt (pfi). EPFI= 1 enables interrupts generated by LBD.
EICON.4	PFI – Power-fail interrupt flag(voltage too low). PFI =1 indicates a power-fail interrupt was detected at the pfi pin. PFI must be cleared by software before exiting the interrupt service routine. Otherwise, the interrupt occurs again. Setting PFI in software generates a power-fail interrupt, if enabled.
EICON.3	/
EICON.2	/
EICON.1	/
EICON.0	/

Table 12 EICONregister

EIE register（0xE8）

bit	definition
EIE.7	EX9： Enable external interrupt 9 EX9 = 0 disables external interrupt 9 (int9_n)

	EX9 = 1 enable interrupts generated by 9(int9_n)
EIE.6	EX8： Enable external interrupt 8 EX8 = 0 disables external interrupt 8 (int8) EX8 = 1 enable interrupts generated by 8 (int8)
EIE.5	EX7： Enable external interrupt 7 EX7 = 0 disables external interrupt 7 (int7_n) EX7 = 1 enable interrupts generated by 7 (int7_n)
EIE.4	EX6： Enable external interrupt 6 EX6 = 0 disables external interrupt 6 (int6) EX6 = 1 enable interrupts generated by 6 (int6)
EIE.3	EX5： Enable external interrupt 5 EX5 = 0 disables external interrupt 5 (int5_n) EX5 = 1 enable interrupts generated by 5 (int5_n)
EIE.2	EX4： Enable external interrupt 4 EX4 = 0 disables external interrupt 4 (int4) EX4 = 1 enable interrupts generated by 4 (int4)
EIE.1	EX3： Enable external interrupt 3 EX3 = 0 disables external interrupt 3 (int3_n) EX3 = 1 enable interrupts generated by 3 (int3_n)
EIE.0	EX2： Enable external interrupt 2 EX2 = 0 disables external interrupt 2 (int2) EX2 = 1 enable interrupts generated by 2 (int2)

Table 13 EIE register

EIP register（0xe9）

bit	definition
EIP.7	PX9： External interrupt 9 priority control. PX9= 0 sets external interrupt 9 to low priority(int9_n) PX9 = 1 sets external interrupt 9 to high priority (int9_n)
EIP.6	PX8： External interrupt 8 priority control. PX8 = 0 sets external interrupt 8 to low priority (int8) PX8 = 1 sets external interrupt 8 to high priority (int8)
EIP.5	PX7： External interrupt 7 priority control. PX7= 0 sets external interrupt 7 to low priority(int7_n) PX7 = 1 sets external interrupt 7 to high priority (int7_n)
EIP.4	PX6： External interrupt 6 priority control. PX6 = 0 sets external interrupt 6 to low priority (int6) PX6 = 1 sets external interrupt 6 to high priority (int6)
EIP.3	PX5： External interrupt 5 priority control. PX5= 0 sets external interrupt 5 to low priority(int5_n) PX5 = 1 sets external interrupt 5 to low priority (int5_n)
EIP.2	PX4： External interrupt 4 priority control. PX4 = 0 sets external interrupt 4 to low priority (int4)

	PX4 = 1 sets external interrupt 4 to high priority (int4)
EIP.1	PX3： External interrupt 3 priority control. PX3= 0 sets external interrupt 3 to low priority(int3_n) PX3 = 1 sets external interrupt 3 to low priority (int3_n)
EIP.0	PX2： External interrupt 2 priority control. PX2 = 0 sets external interrupt 2 to low priority (int2) PX2 = 1 sets external interrupt 2 to high priority (int2)

Table 14 EIP register

6 Reset system

There are three active low reset source in BK5100, they are power on reset, reset pin, watch dog reset. After reset, the MCU will re-start from address 0.

7 Clock system

There are three clock domains in BK5100, clock for MCU, clock for USB, and clock for RF part.

48MHz clock is used by USB, and you can shut down USB clock for power saving by setting SFR register USB_PWR_CN.1.

The clock frequency for RF module is always 16MHz.

There are two sources for MCU clock, one is XOSC16M, and another is RCOSC32KHz.

32KHz is used for sleep mode. There are two 32k source in the chip, one is divided from 16M, and another is generated by RC circuit. The previos clock source is more accurate than the RC 32k clock, but need more power.

16MHz is used for normal work mode. In normal work mode, the clock for MCU is divided by 16MHz clock source, it can be set by setting register CKCON[7:6]. (SFR 0X8E)

00： 16Mhz

01： 8Mhz

10： 4Mhz

11： 2Mhz

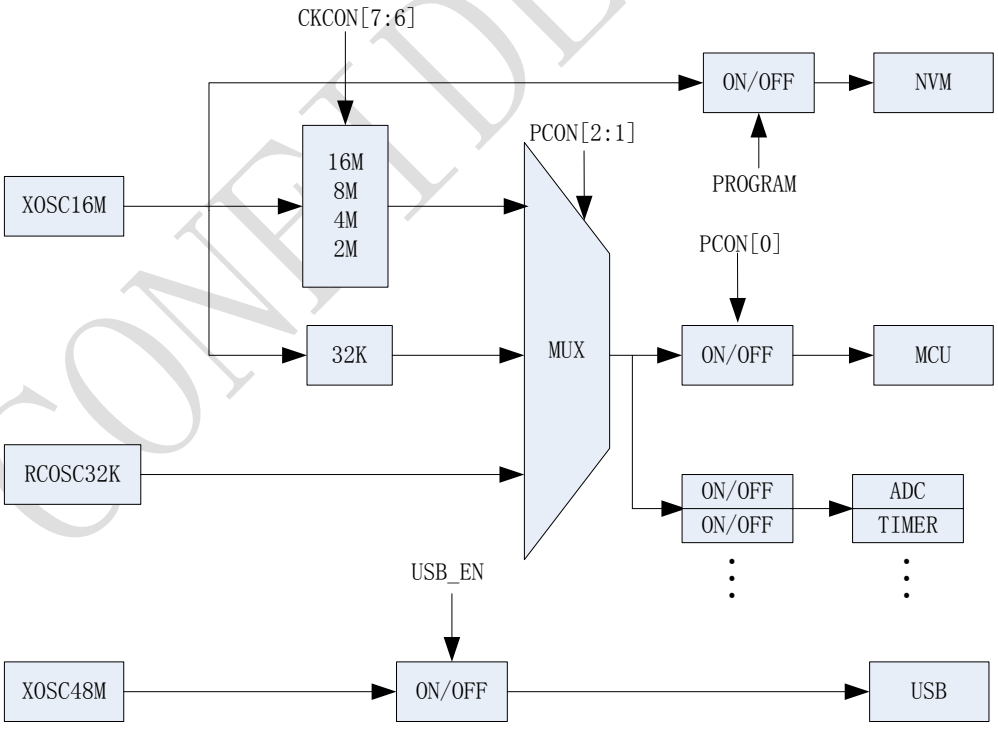


Figure 11 Clock System

8 General I/O

8.1 Basic function

There are four general ports P0, P1, P2 in BK5100. All the three ports can be used for general I/O with selectable direction for each bit, or these lines can be used for specialized functions.

When the four ports are configured as general I/O, the detail function of them can be set by twenty SFR register described next.

Note: the default setting for all the ports is general I/O.

SFR Pn (n=0-2): general IO data register

SFR Pn_IOSEL (n=0-2): input/output select register

1： INPUT (default)

0：OUPUT

Note: this bit is valid only when the port is configured as general IO port

SFR Pn_OPDR (n=0-2); open_drain select register

1：open-drain，(default)

0：Not open-drain

Note: this bit is valid only when the port is configured as general IO port

SFR Pn_PU (n=0-2); pull up enable or not

1: pull up enable

0: pull up disable (default)

SFR Pn_PD (n=0-2): pull down enable or not

1: pull down enable (default)

0: pull down disable

SFR Pn_WUEN (n=0-2); wake up enable register

1：wake up enable，

0：wake up disable (default)

Note: this bit is valid only when the port is configured as general IO port

8.2 Wake up function

SFR Pn_WUEN (n=0-2); wake up enable register

1：wake up enable，

0：wake up disable (default)

SFR Pn_WUMOD (n=0-2); wake up enable register （SFR E1~E3）

1：wake up by rise or fall edge，

0：wake up by fall edge (default)

Note: this bit is valid only when the port is configured as input .

8.3 Second function of port

8.3.1 Sencond function

When one port is configured as second function, the direction (input or output) is decided by the second function and is irrespective of Pn_IOSEL.

Port0	7	6	5	4	3	2	1	0
Second function					T2EX	t2	t1	t0

Table 15 PORT0 Second Function

T0: Timer0 external input (when PALT.6 ==1)
T1: Timer1 external input (when PALT.5 ==1)
T2: Timer2 external input (when PALT.4 ==1)
T2EX: timer2 external capture/reload trigger inputPALT.4 ==1)

Port1	7	6	5	4	3	2	1	0
Second function	/	/	/	/	/	/	int1_n	int0_n

Table 16 PORT1 Second Function

Int0_n : external interrupt 0 input pin, active low (when EA&IE0 == 1)
Int1_n : external interrupt 1 input pin, active low (when EA&IE1 == 1)

Port2	7	6	5	4	3	2	1	0
Second function			/			rxd_out	txd	rxd_in

Table 17 PORT2 Second Function

rxd_in : the input for UART (when (EA & ES0) || PALT.7 ==1)
txd : the output for UART (when (EA & ES0) || PALT.7 ==1)
rxd_out : the output pin pin for received data, only used for mode0 (when(EA & E S0) || PALT.7 ==1)

The second function of port is enabled when the corresponding interrupt is enabled or the corresponding bit of register PALT is valid. (When the port is used as second function, IOSEL is invalid.)

For example:

When SETB EA
SETB ET0

The PORT0.0 is configured as the input timer0. Also, you can set PALT |= 0x40 to get the same result.

Note: EA and ET0 should be valid at the same time, or the corresponding port is general IO still.

8.3.2 SECOND FUCNTION FOR GPIO

port	No.	Basic Function1	Second Function
P0.0	1	GPIO	Timer0
P0.1	2	GPIO	Timer1
P0.2		GPIO	Timer2
P0.3		GPIO	NC
P0.4	3	GPIO	NC
P0.5	4	GPIO	
P0.6	5	GPIO	
P0.7	6	GPIO	
P1.0	7	GPIO	EXT INT
P1.1	8	GPIO	
P1.2		GPIO	NC
P1.3		GPIO	NC
P1.4		GPIO	NC
P1.5		GPIO	NC
P1.6		GPIO	NC
P1.7		GPIO	NC
P2.0	10	GPIO	UART
P2.1	11	GPIO	
P2.2			
P2.3	12	GPIO	NC
P2.4	13	GPIO	
			NC
			NC
P2.7	13	GPIO	NC

Table 18 GPIO for 32-pin package

9 Watch Dog

There is a watch dog timer in BK5100. When overflow happened, the WDT will trigger the CPU into reset status and rerun from the beginning location. The software need feed the dog termly to avoid the overflow happen.

Note: the reset does not affect RF part.

Ther are two methods to enable the WDT.

One is writing 0Xa5 on SFR address 0x97(WDCON) and this operation will clear the WDT counter also (feed dog). Once the WDT enabled by this method, you can disable the WDT through writing 0XDE and 0XAD consecutively during eight clock periods. When the WDT enabled by this method, you can also set whether running in IDLE state. To do this, you can enable it by writing 0XD1 on SFR address 0x97 or disable it by writing 0XDE and 0XDA consecutively during eight clock periods.

The other method is writng 0XFF on SFR address 0x97. You can not close it once enable WDT through this method except any reset happened. In this status, the WDT will run always even in IDLE state.

WDCON	7	6	5	4	3	2	1	0
0x97	/	/	/	state	/	ps2	ps1	ps0

Table 19 Watch Dog Register

state: read only 1: the WDT in active status 0: the WDT in inactive status

Ps2, ps1, ps0: the prescale of watch dog clock.

PS2	PS1	PS0	PRE_scale
0	0	0	2
0	0	1	4
0	1	0	8
0	1	1	16
1	0	0	32
1	0	1	64
1	1	0	128
1	1	1	256

Table 20 The Prescale of Watch Dog clock

The overflow time of watch dog:

$$WatchdogOverflowTime = \frac{PRE\_scale \times 32768}{\text{system clock}}$$

When overflow occur, the whole system will be reset.

10 Timer

The BK51 includes three timer/counters (Timer 0 and Timer 1 Timer2). Each timer/counter can operate as either a timer with a clock rate based on the clk input, or as an event counter clocked by the t0 pin (Timer 0), t1pin (Timer 1) or the t2 pin (Timer 2).

10.1 Timer0 and Time1

Timers 0 and 1 each operate in four modes, as controlled through the TMOD SFR and the TCON SFR. The four modes are:

-13-bit timer/counter (mode 0)
-16-bit timer/counter (mode 1)
-8-bit counter with auto-reload (mode 2)
-Two 8-bit counters (mode 3, Timer 0 only)

The next is the register description：

Bits	Definition
TMOD.7	GATE – Timer1 gate control. When GATE=1, Timer1 will clock only when int1_n= 1 and TR1 (TCON.6) =1. When GATE=0, Timer1 will clock only when TR1= 1, regardless of the state of int1_n.
TMOD.6	C/T – Counter/Timer select. When C/T=0, Timer 1 is clocked by clk/4 or clk/12, depending on the state of T1M (CKCON.4). When C/T=1, Timer1 is clocked by the t1(P0.1) pin.
TMOD.5	M1：Timer1 mode selects bit1.
TMOD.4	M0: Timer1 mode selects bit0. M1M0: 00 mode0 01 mode1 10 mode2 11 mode3
TMOD.3	GATE – Timer 0 gate control. When GATE=1, Timer 0 will clock only when int0_n= 1 and TR0 (TCON.4)=1. When GATE= 0, Timer 0 will clock only when TR0= 1, regardless of the state of int0_n.
TMOD.2	C/T – Counter/Timer select. When C/T=0, Timer 0 is clocked by clk/4 or clk/12, depending on the state of T0M (CKCON.3). When C/T=1, Timer 0 is clocked by the t0 pin
TMOD.1	M0: Timer0 mode select bit1.
TMOD.0	M0: Timer0 mode select bit0. M1M0: 00 mode0 01 mode1 10 mode2 11 mode3

Table 21 Timer TMOD Register

Bits	Definition
TCON.7	TF1 – Timer1 overflow flag. Set to1 when the Timer1 count overflows and cleared when the CPU vectors to the interrupt service routine.
TCON.6	TR1 – Timer1 run control. Set to1 to enable counting on Timer1.
TCON.5	TF0 – Timer 0 overflow flag. Set to1 when the Timer 0 count overflows and cleared when the CPU vectors to the interrupt service routine.
TCON.4	TR0 – Timer 0 run control. Set to1 to enable counting on Timer 0.(auto clear)
TCON.3	IE1 – Interrupt 1 detect. IE1 is set by hardware when an interrupt is detected on the int1_n pin and is automatically cleared when the CPU vectors to the corresponding interrupt service routine.
TCON.2	IT1 – Interrupt 1 type select. When IT1=1, the BK51 detects int1_n on the falling edge (edge-sensitive). When IT1=0, the BK51 detects int1_n as a low level (level-sensitive).
TCON.1	IE0 – Interrupt 1 detect. IE0 is set by hardware when a interrupt is detected on the int0_n pin and is automatically cleared when the CPU vectors to the corresponding interrupt service routine.
TCON.0	IT0 – Interrupt 0 type select. When IT0=1, the BK51 detects int0_n on the falling edge (edge-sensitive). When IT0=0, the BK51 detects int0_n as a low level (level-sensitive).

Table 22 Timer TCON Register

10.1.1 Mode0

Mode 0 operation, illustrated in next Figure, is the same for Timer 0 and Timer 1. In mode 0, the timer is configured as a 13-bit counter that uses bits0–4 of TL0 (or TL1) and all 8 bits of TH0 (or TH1). The timer enable bit (TR0/TR1) in the TCON SFR starts the timer. The C/T bit selects the timer/counter clock source, MCU clk or t0/t1 pin.

When the 13-bit count increments from1FFFh (all ones), the counter rolls over to all zeros, the TF0 (or TF1) bit is set in the TCON SFR.

10.1.2 Mode1

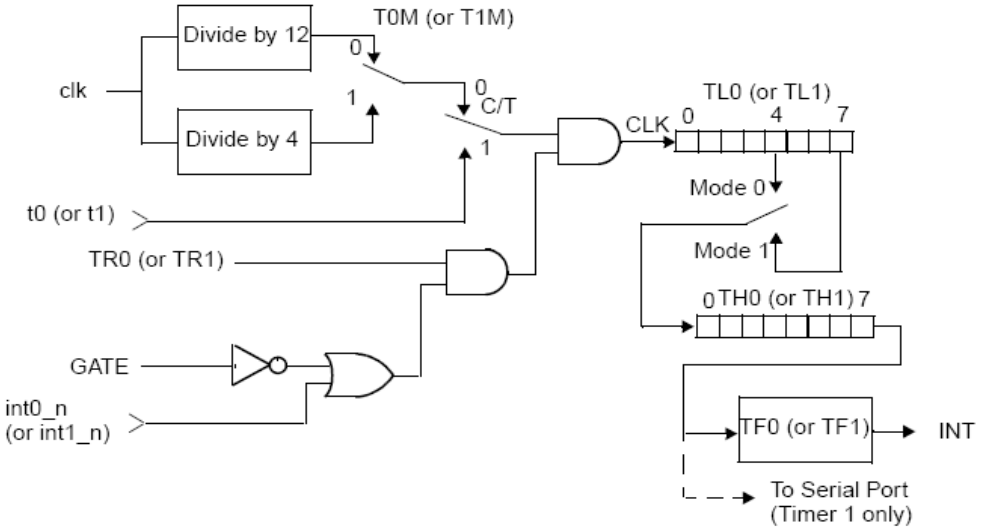


Figure 12 Timer 0/1 – Modes 0 and 1

Mode 1 operation is the same for Timer 0 and Timer 1. In mode 1, the timer is configured as a 16-bit counter. All 8 bits of the LSB register (TL0 or TL1) are used. The counter rolls over to all zeros when the count increments from FFFFh. Otherwise, mode 1 operation is the same as mode 0.

10.1.3 Mode2

Mode 2 operation is the same for Timer 0 and Timer 1. In mode 2, the timer is configured as an 8-bit counter, with automatic reload of the start value. The LSB register (TL0 or TL1) is the counter and the MSB register (TH0 or TH1) stores the reload value.

Mode 2 counter control is the same as for mode 0 and mode 1. However, in mode 2, when TLn increments from FFh, the value stored in THn is reloaded into TLn.

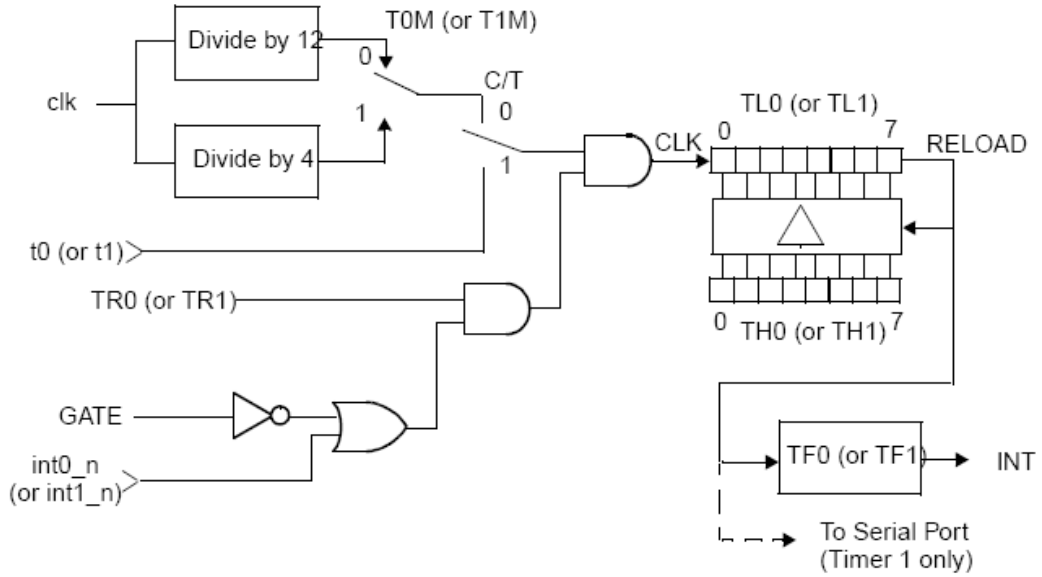


Figure 13 Timer 0/1 – Mode 2

10.1.4 Mode3

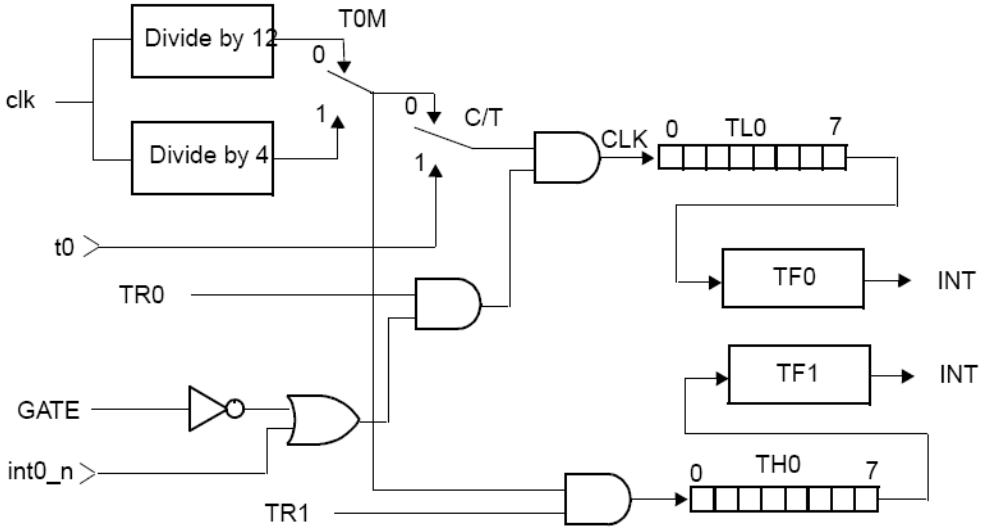


Figure 14 Timer 0 – Mode 3

In mode 3, Timer 0 operates as two 8-bit counters and Timer 1 stops counting and holds its value.

As shown in above figure, TL0 is configured as an 8-bit counter controlled by the normal Timer 0 control bits. TL0 can either count clk cycles (divided by 4 or by 12) or high-to-low transitions on t0, as determined by the C/T bit. The GATE function can be used to give counter enable control to the int0_n signal.

TH0 functions as an independent 8-bit counter. However, TH0 can only count clk

cycles (divided by 4 or by12). The Timer 1 control and flag bits (TR1 and TF1) are used as the control and flag bits for TH0.

When Timer 0 is in mode 3, Timer 1 has limited usage because Timer 0 uses the Timer 1 control bit (TR1) and interrupts flag (TF1). Timer 1 can still be used for baud rate generation and the Timer 1 count values are still available in the TL1 and TH1 registers.

10.2 Timer – Rate Conrol

Using the default rate (12 clocks per timer increment) allows existing application code with real-time dependencies, such as baud rate, to operate properly. However, applications that require fast timing can set the timers to increment every 4clk cycles by setting bits in the Clock Control register (CKCON) at SFR.

CKCON.5 control Timer2 clock rate
CKCON.4 control Timer1 clock rate
CKCON.3 control Timer0 clock rate

When a CKCON register bit is set to1, the associated counter increments at 4-clk intervals. When a CKCON bit is cleared, the associated counter increments at 12-clk intervals. The timer controls are independent of each other. The default setting for all the three timers is 0 (12-clk intervals).

Note: when setting timer 2 as Baud-Rate generator, the frequency is half of CPUclock, and regardless of the value in CKCON.5.

10.3 Timer2

Timer 2 runs only in 16-bit mode and offers several capabilities not available with Timers 0 and1. The modes available with Timer 2 are:

- 16-bit timer/counter
- 16-bit timer with capture
- 16-bit auto-reload timer/counter
- Baud rate generator

The SFRs associated with Timer 2 are:

T2CON

Bits	Definition
T2CON.7	TF2 – Timer 2 overflow flag. Hardware will set TF2 when Timer 2 overflows from FFFFh. TF2 must be cleared to 0 by the software. TF2 will only be set to a 1 if RCLK and TCLK are both cleared to 0. Writing a1 to TF2 forces a Timer 2 interrupt if enabled.
T2CON.6	EXF2 – Timer 2 external flag. Hardware will set EXF2 when a reload or capture is caused by a high-to-low transition on the t2ex pin, and EXEN2 is set. EXF2 must be cleared to 0 by the software. Writing a 1 to EXF2 forces a Timer 2 interrupt if enabled.
T2CON.5	RCLK – Receive clock flag. Determines whether Timer1 or Timer 2 is used for Serial Port 0 timing of received data in serial mode 1 or 3. RCLK =1 selects Timer 2 overflow as the receive clock. RCLK=0 selects Timer1

	overflow as the receive clock.
T2CON.4	TCLK – Transmit clock flag. Determines whether Timer 1 or Timer 2 is used for Serial Port 0 timing of transmit data in serial mode 1 or 3. TCLK =1 selects Timer 2 overflow as the transmit clock. TCLK=0 selects Timer1 overflow as the transmit clock.
T2CON.3	EXEN2 – Timer 2 external enable. EXEN2=1 enables capture or reload to occur as a result of a high-to-low transition on t2ex, if Timer 2 is not generating baud rates for the serial port. EXEN2= 0 causes Timer 2 to ignore all external events at t2ex.
T2CON.2	TR2 – Timer 2 run control flag. TR2=1 starts Timer 2. TR2=0 stops Timer 2.
T2CON.1	C/T2 – Counter/timer select. C/T2= 0 selects a timer function for Timer 2. C/T2=1 selects a counter of falling transitions on the t2 pin. When used as a timer, Timer 2 runs at 4 clocks per increment or12 clocks per increment as programmed by CKCON.5, in all modes except baud rate generator mode. When used in baud rate generator mode, Timer 2 runs at 2 clocks per increment, independent of the state of CKCON.5.
T2CON.0	CP/RL2 – Capture/reload flag. When CP/RL2= 1, Timer 2 captures occur on high-to-low transitions of t2ex, if EXEN2= 1. When CP/RL2=0, auto-reloads occur when Timer 2 overflows or when high-to-low transitions occur on t2ex, if EXEN2= 1. If either RCLK or TCLK is set to1, CP/RL2 will not function and Timer 2 will operate in auto-reload mode following each overflow.

Table 23Timer2 T2CON Register

Next table summarizes how the SFR bits determine the Timer 2 mode.

RCLK ,TCLK	CP/RL2	TR2	Mode
00	0	1	16-bit timer/counter with auto-reload
00	1	1	16-bit timer/counter with capture
1x or x1	x	1	Baud rate generator
xx	x	0	off

Table 24 Timer 2 Mode Control Summary

RCAP2L: Used to capture the TL2 value when Timer 2 is configured for capture mode, or as the LSB of the 16-bit reload value when Timer 2 is configured for auto-reload mode.

RCAP2H: Used to capture the TH2 value when Timer 2 is configured for capture mode, or as the MSB of the 16-bit reload value when Timer 2 is configured for auto-reload mode.

TL2: Lower 8 bits of the 16-bit count.

TH2: Upper 8 bits of the 16-bit count.

10.3.1 16-Bit Timer/Counter Mode

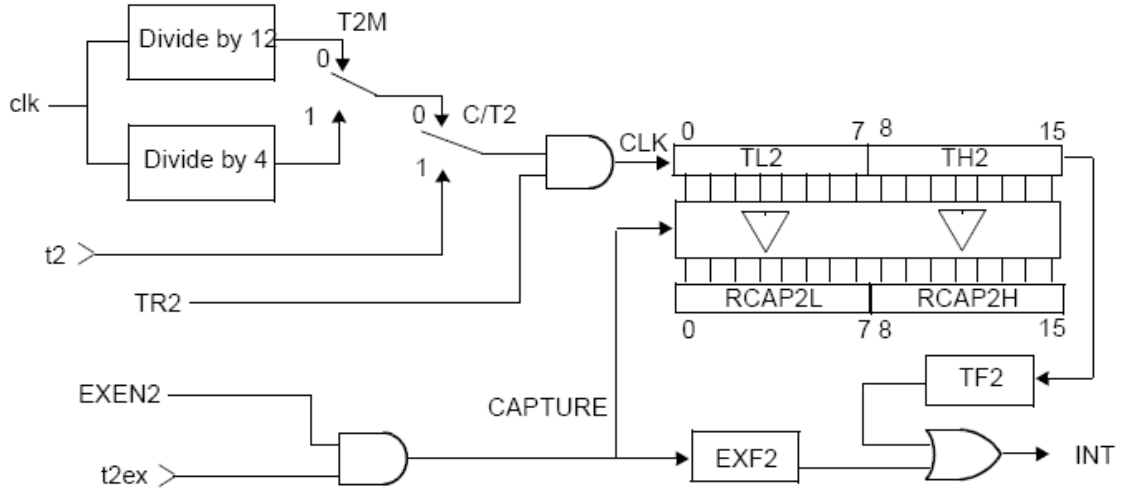


Figure 15 Timer 2 – Timer/Counter with Capture

The above figure illustrates how Timer 2 operates in timer/counter mode with the optional capture feature. The C/T2 bit determines whether the 16-bit counter counts clk cycles (divided b 4 or 12), or high-to-low transitions on the t2 pin. The TR2 bit enables the counter. When the count increment from FFFFh, the TF2 flag is set and t2_out goes high for one clk cycle.

The Timer 2 capture mode is the same as the 16-bit timer/counter mode, with the addition of the capture registers and control signals. The CP/RL2 bit in the T2CON SFR enables the capture feature. When CP/RL2 = 1, a high-to-low transition on t2ex when EXEN2= 1 causes the Timer 2 value to be loaded into the capture registers (RCAP2L and RCAP2H).

10.3.2 16-Bit Timer/Counter Mode with Auto-Reload

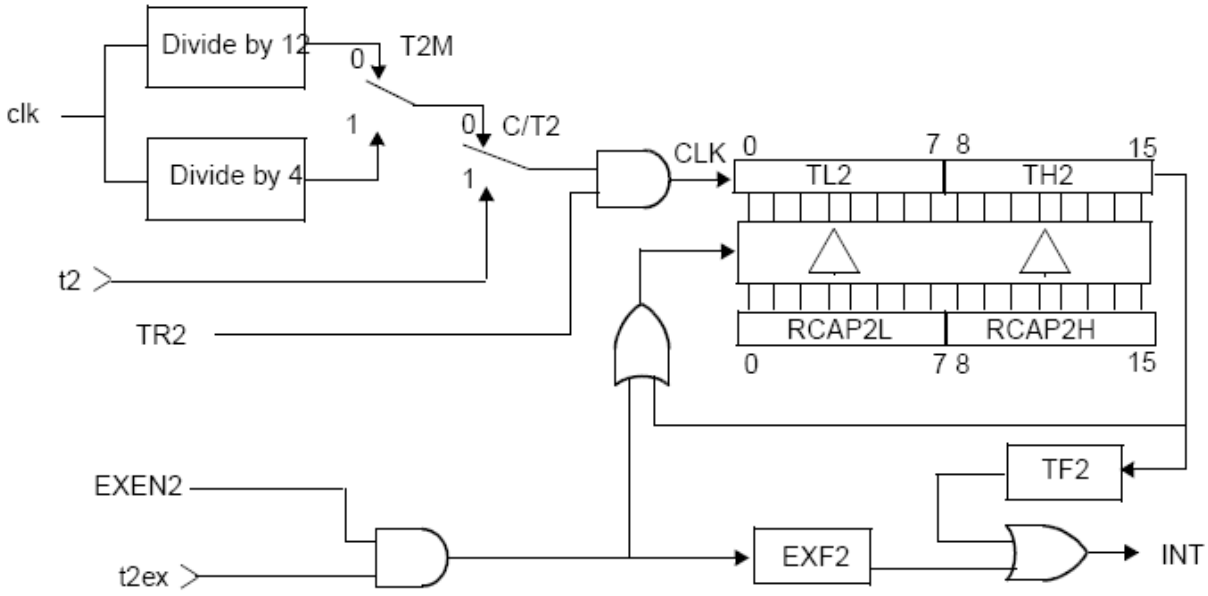


Figure 16 Timer 2 – Timer/Counter with Auto-Reload

When CP/RL2= 0, Timer 2 is configured for the auto-reload mode illustrated in above figure . Control of counter input is the same as for the other 16-bit counter modes. When the count increments from FFFFh, Timer 2 sets the TF2 flag and the starting value is reloaded into TL2 and TH2. The software must preload the starting value into the RCAP2L and RCAP2H registers.

When Timer 2 is in auto-reload mode, a reload can be forced by a high-to-low transition on the t2ex pin, if enabled by EXEN2 = 1.

10.3.3 Baud Rate Generator Mode

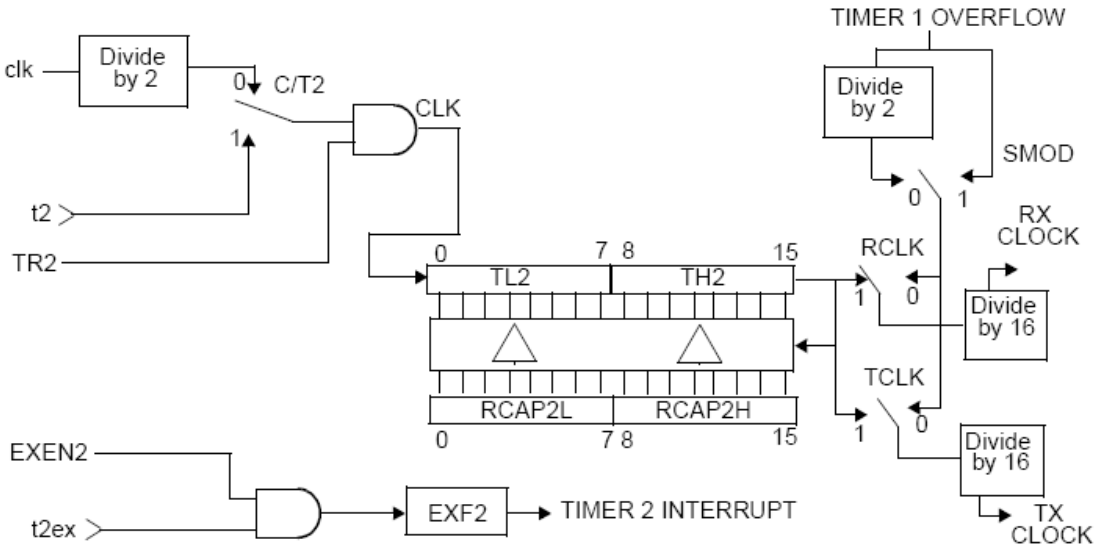


Figure 17 Timer 2 – Baud Rate Generator Mode

Setting either RCLK or TCLK to1 configures Timer 2 to generate baud rates for Serial Port in serial mode 1 or 3. In baud rate generator mode, Timer 2 functions in autoreload mode. However, instead of setting the TF2 flag, the counter overflow generates a shift clock for the serial port function. As in normal auto-reload mode, the overflow also causes the preloaded start value in the RCAP2L and RCAP2H registers to be reloaded into the TL2 and TH2 registers.

When either TCLK = 1 or RCLK = 1, Timer 2 is forced into auto-reload operation, regardless of the state of the CP/RL2 bit.When operating as a baud rate generator, Timer 2 does not set the TF2 bit. In this mode, a Timer 2 interrupt can only be generated by a high-to-low transition on the t2ex pin setting the EXF2 bit, and only if enabled by EXEN2 = 1.

The counter time base in baud rate generator mode is clk/2. To use an external clock source, set C/T2 to 1 and apply the desired clock source to the t2 pin.

11 UART

Serial Port of BK51 is identical in operation to the standard 8051 serial port. The data sent includes start bit, data bits (LSB send first) and stop bit at least.

The register inculdes：

SBUF: Serial port R/W register

SCON：Serial port control register

bits	Description
SCON.7	SM0-Serial port mode bit0.
SCON.6	SM1-Serial port mode bit 1, decoded as: SM0-SM1 mode 0 0 0 0 1 1 1 0 2 1 1 3
SCON.5	SM2 – Multiprocessor communication enable. In modes 2 and 3, SM2 enables the multiprocessor communication feature. If SM2=1 in mode 2 or 3, RI will not be activated if the received 9th bit is 0. If SM2=1 in mode 1, RI will only be activated if a valid stop is received.
SCON.4	REN – Receive enable. When REN=1, reception is enabled.
SCON.3	TB8 – Defines the state of the 9th data bit transmitted in modes 2 and 3.
SCON.2	RB8 – In modes 2 and 3, RB8 indicates the state of the 9th bit received. In mode1, RB8 indicates the state of the received stop bit. In mode 0, RB8 is not used.
SCON.1	TI – Transmit interrupt flag. Indicates that the transmit data word has been shifted out. In mode 0, TI is set at the end of the 8th data bit. In all other modes, TI is set when the stop bit is placed on the txd0 pin. TI must be cleared by the software.
SCON.0	RI – Receive interrupt flag. Indicates that a serial data word has been received. In mode 0, RI_1 is set at the end of the 8th data bit. If serial port interrupt enabled(EA=1 and ES=1), it will generate interrupt when RI=1.

Table 25 UART SCON Register

11.1 Mode0

Serial mode 0 provides synchronous, half-duplex serial communication.

The serial mode 0 baud rate is either clk/12 or clk/4, depending on the state of the SM2 bit. When SM2= 0, the baud rate is clk/12; when SM2 = 1, the baud rate is clk/4.

Mode 0 operation is identical to the standard 8051. Data transmission begins when an instruction writes to the SBUF SFR. The UART shifts the data out, LSB first, at the

selected baud rate, until the 8-bit value has been shifted out.

Mode 0 data reception begins when the REN bit is set and the RI bit is cleared in the corresponding SCON SFR. The shift clock is activated and the UART shifts data in on each rising edge of the shift clock until 8 bits have been received. One machine cycle after the 8th bit is shifted in, the RIbit is set and reception stops until the software clears the RI bit.

The next two figure illustrate Serial Port Mode 0 transmit and receive timing.

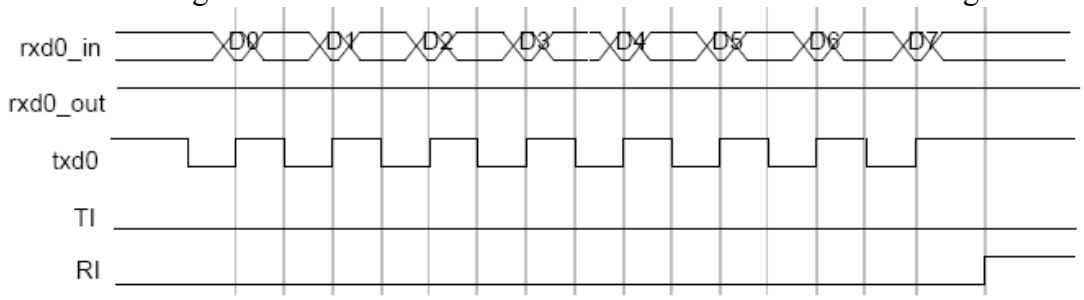


Figure 18 Uart Mode0 Tramsmit Timing

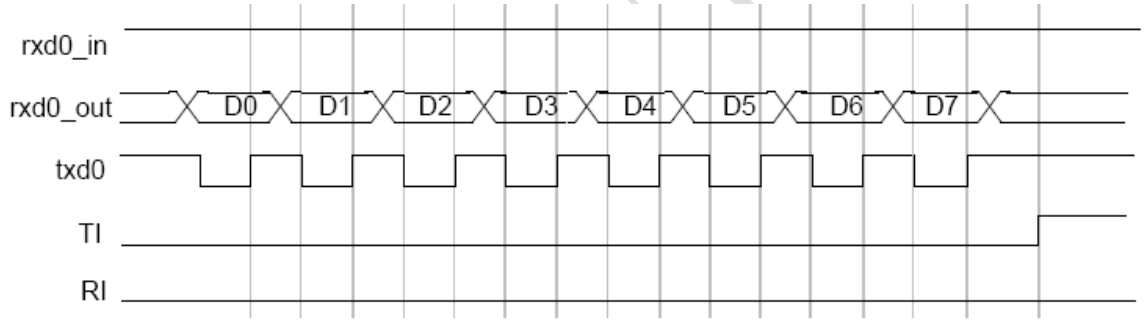


Figure 19 Uart Mode0 Receive Timing

11.2 Mode1

When SM=01, serial port works as mode 1. Mode 1 provides standard asynchronous, full-duplex communication, using a total of 10 bits: 1 start bit, 8data bits, and 1 stop bit. For receive operations, the stop bit is stored in RB8. Data bits are received and transmitted LSB first.

11.2.1 Mode1 Baud Rate

The mode 1 baud rate is a function of timer overflow. Serial Port can use either Timer 1 or Timer 2 to generate baud rates.

Each time the timer increments from its maximum count (FFh for Timer 1 or FFFFh for Timer 2), a clock is sent to the baud rate circuit. The clock is then divided by16 to generate the baud rate.

(1) using Timer1

When using Timer 1, the SMOD (the MSB bit of PCON) bit selects whether or not to divide the Timer 1 rollover rate by 2. Therefore, when using Timer 1, the baud rate is determined by the equation:

$$Baud-Rate=\frac{2^{SMOD}}{32}\times \text{Timer1 Overflow}$$

To use Timer 1 as the baud rate generator, it is best to use Timer 1 mode 2 (8-bit counter with auto-reload), although any counter mode can be used. The Timer 1 reload value is stored in the TH1 register, which makes the complete formula for Timer 1:

$$Baud-Rate=\frac{2^{SMOD}}{32}\times\frac{clk}{12\times(256-TH1)}$$

The 12 in the denominator in the above equation can be changed to 4 by setting the T1M bit in the CKCON SFR. To derive the required TH1 value from a known baud rate (when TM1 = 0), use the equation:

$$TH1=256-\frac{2^{SMOD}\times clk}{384\times Baud-Rate}$$

(2) When using Timer2

When using Timer 2, the baud rate is determined by the equation:

$$Baud-Rate=\frac{\text{Timer2 Overflow}}{16}$$

To use Timer 2 as the baud rate generator, configure Timer 2 in auto-reload mode and set the TCLK and/or RCLK bits in the T2CON SFR. TCLK selects Timer 2 as the baud rate generator for the transmitter; RCLK selects Timer 2 as the baud rate generator for the receiver. The 16-bit reload value for Timer 2 is stored in the RCAP2L and RCA2H SFRs, which makes the equation for the Timer 2 baud rate:

$$Baud-Rate=\frac{clk}{32\times(65536-RCAP2H,RCAP2L)}$$

Where RCAP2H,RCAP2L is the content of RCAP2H and RCAP2L taken as a 16-bit unsigned number.

To derive the required RCAP2H and RCAP2L values from a known baud rate, use the equation:

$$RCAP2H,RCAP2L=65536-\frac{clk}{32\times Baud-Rate}$$

The 32 in the denominator is the result of the clk being divided by2 and the Timer 2 overflow being divided by16.

11.2.2 Mode1 tramsmit

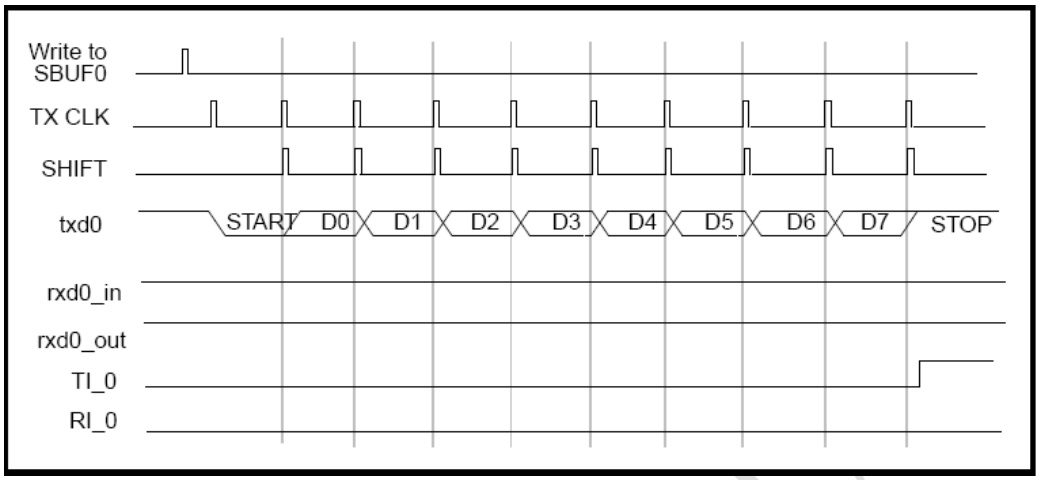


Figure 20 Uart Mode1 Transmit Timing

In mode 1, the UART begins transmitting after the first rollover of the divide-by-16 counter after the software writes to the SBUF0 (or SBUF1) register. The UART transmits data on the txd pin in the following order: start bit, 8data bits (LSB first), stop bit. The TIbit is set 2 clk cycles after the stop bit is transmitted.

11.2.3 Mode1 Receive

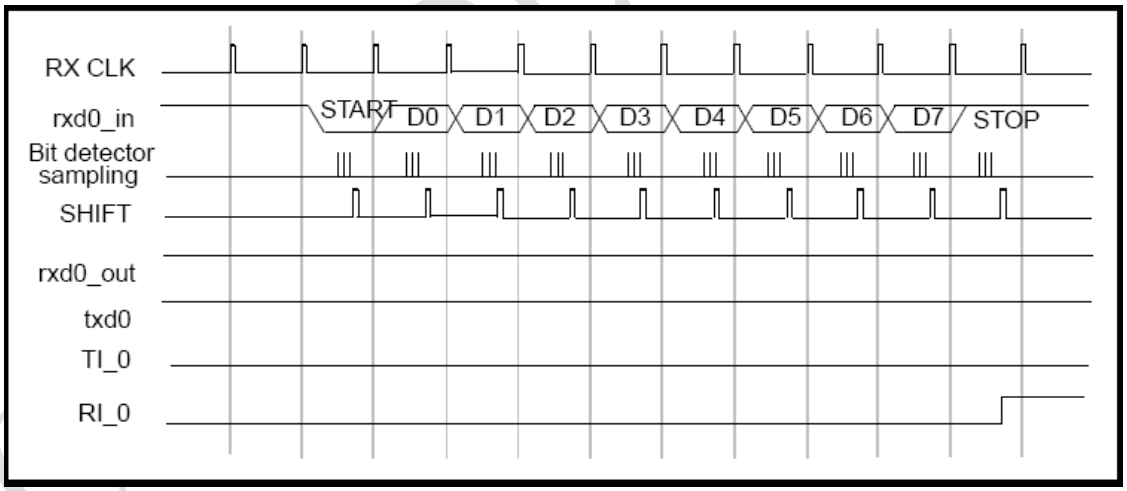


Figure 21 Uart Mode1Receive Timing

Reception begins at the falling edge of a start bit received on rxd_in, when enabled by the REN bit. For this purpose, rxd_in is sampled sixteen times per bit for any baud rate. When a falling edge of a start bit is detected, the divide-by-16 counter used to generate the receive clock is reset to align the counter rollover to the bit boundaries.

For noise rejection, the serial port establishes the content of each received bit by a majority decision of three consecutive samples in the middle of each bit time. This is especially true for the start bit. If the falling edge on rxd_in is not verified by a majority decision of three consecutive samples (low), then the serial port stops reception and waits

for another falling edge on rxd_in.

At the middle of the stop bit time, the serial port checks for the following conditions:

- RI= 0,
- and If SM2= 1, the state of the stop bit is1.

(If SM2= 0, the state of the stop bit doesn't matter.)

If the above conditions are met, the serial port then writes the received byte to the SBUF register, loads the stop bit into RB8, and sets the RI bit. If the above conditions are not met, the received data is lost, the SBUF register and RB8 bit are not loaded, and the RI bit is not set. After the middle of the stop bit time, the serial port waits for another high-to-low transition on the (rxd_in) pin.

11.3 Mode2

Mode 2 provides asynchronous, full-duplex communication, using a total of 11bits:

- One start bit
- Eight data bits
- One programmable 9th bit
- One stop bit.

The data bits are transmitted and received LSB first. For transmission, the 9thbit is determined by the value in TB8. To use the 9th bit as a parity bit, move the value of the P bit (SFR PSW.0) to TB8.

Transmission begins after the first rollover of the divide-by-16 counter following a software write to SBUF. The UART shifts data out on the txd pin in the following order: start bit, data bits (LSB first), 9th bit, stop bit. The TI bit is set when the stop bit is placed on the txd pin.

Reception begins at the falling edge of a start bit received on rxd_in, when enabled by the RENbit. For this purpose, rxd_in is sampled sixteen times per bit for any baud rate. When a falling edge of a start bit is detected, the divide-by-16 counter used to generate the receive clock is reset to align the counter rollover to the bit boundaries.

The next figure is the timing of receive and transmit in mode 2.

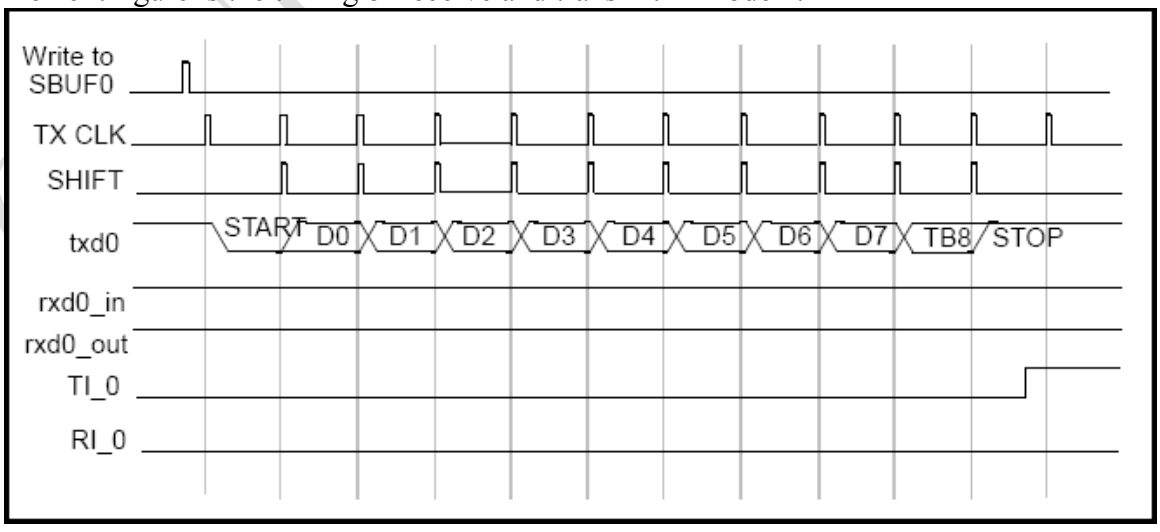


Figure 22 Uart Mode2 Transmit Timing

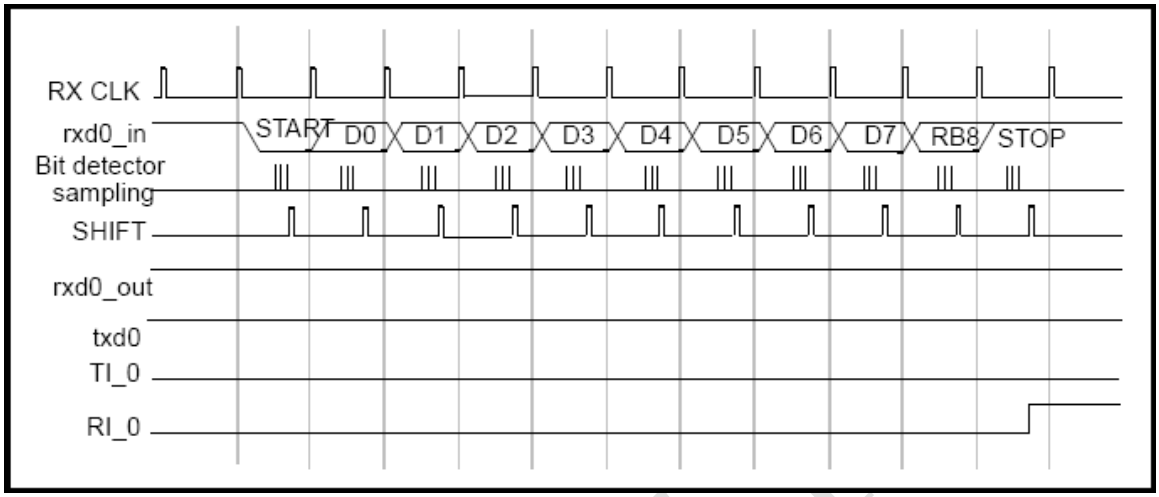


Figure 23 Uart Mode2 Receive Timing

11.4 Mode3

Mode 3 provides asynchronous, full-duplex communication, using a total of 11bits:

- One start bit
- Eight data bits
- One programmable 9th bit
- One stop bit. The data bits are transmitted and received LSB first.

The mode 3 transmits and operations are identical to mode 2. The mode 3 baud rate generation is identical to mode 1. That is, mode 3 is a combination of mode 2 protocol and mode 1 baud rate.

12 LBD

The LBD circuit is used to monitor power supply.

SFR	7-2	1	0
0xD8	reserved (dont change)	LBD_thd[4]	reserved (dont change)

Table 26 LBD register 1

SFR	7 - 4	3-0
0xD9	reserved (dont change)	LBD_thd[3:0]

Table 27 LBD register 2

SFR	7	6	5-0
0xDA	reserved (dont change)	LBD power on/off	reserved (dont change)

Table 28 LBD register 3

LBD_thd	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
voltage	0.75	0.82	0.89	0.97	1.04	1.12	1.19	1.26	1.34	1.41	1.49	1.55	1.64	1.71	1.79	1.86

LBD_thd	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
voltage	1.95	2.02	2.09	2.16	2.24	2.31	2.39	2.46	2.53	2.6	2.68	2.76	2.83	2.9	2.97	3

Table 29 LBD votage table

The recommend detect process is described as follow:

1. Config the LBD_THD register correctly according to **Table 29**
2. Power up LBD circuit (write 1 into bit 6 of adress 0xDA.)
3. Wait about 50us
4. Clear register EICON.4
5. Read register EICON.4

   If EICON.4=1 it means that that POWER FAIL happened. (LBD voltage is less than the LBD_THD votage now)
6. Power down LBD circuit. (write 0 into bit 6 of adress 0xDA.)

13 Random Number Generator

There are two random number generators in BK5100. One is pseudo random number generator and the other is a true random number generator (RNG), which uses thermal noise to produce a non-deterministic bit stream. The bits are then queued into an 8-bit register for parallel readout. You can use either of them as you selected.

The RNG is interfaced through two registers; RNG_CTL and RNGDAT. RNG_CTL contains two control bits and a status bit. RNGDAT contains the random data.

RNG_CTL	7	6	5	4-0
0xEC	RNG_pwd (RW)		RNG_rdy (R) Used by true RNG only	/

RNG_pwd: write 0 to power up the true RNG module, 1 to power off the true RNG and use the pseudo RNG module.

RNG_rdy: read only register, when RNGDAT is ready RNG_rdy is set to1 by hardware, when software read RNGDAT, it's clear by hardware. Used by true RNG only.

RNG_DAT	7-0
0xEB	Data/seeds

RNG_DAT: RNG_rdy=1 to indicate the true random number (RNG_DAT) is ready for read.

When use pseudo random generator, RNG_DAT is always ready in the register, you can read the result at any time.

Also, you can change the seed by writing to SFR 0xEB. A different seed can be stored in the MTP in advance to get a different random number with different chip. For software, the seed should be read out from MTP and write to the SFR when power on.

When use the true random generator, it will take about 50us to prepare the first random number after you power up the module. After that, it will take about 2us to generate a new 8-bits random number.

14 USB2.0

The USB module in BK5100 provides a full speed USB function interface that meets the 1.1 and 2.0 specification. USB module has 8 endpoints and the depth and start address of every endpoint FIFO can be configured. The FIFO can locate any position of the 2K EXRAM. It supports control, interrupt, bulk, synchronous transfer mode; also it supports multiple-buffer operation controlled by software for using the USB bandwidth sufficiently.

Note: It is assumed the reader is familiar with or has access to the supporting documents USB1.1.

14.1 Clock

USB clock is 48MHz which is generated by PLL integrated in the chip. USB module can enter into idle mode for saving power consumption by setting the USB_PWR_CN.1 (0x0841) SFR. In this state, the register of USB can be read or write, but the USB engine is halted and cannot respond any external operation.

14.2 USB Register Access

The register of USB located from 0x0808 to 0x0850 of external RAM. The access method is same to external RAM, use MOVX command.

USB register included interrupt register, configure register, power management register and address register.

14.3 ENDPOINT Configuration

The USB module should be configured before using USB to communicate. The configure item includes endpoint address in EXRAM, the depth of FIFO, how many endpoints are used, and the direction, mode, enable of every endpoints.

Next is the description of these register. All the register can read or write by software.

EP_ADDR_MSB [0x0840]
CFG_EP0_1 [0x0810], CFG_EP0_0 [0x0811] (endpoint 0 configure register**)**
CFG_EP1_1 [0x0812], CFG_EP1_0 [0x0813] (endpoint 1 configure register**)**
CFG_EP2_1 [0x0814], CFG_EP2_0 [0x0815] (endpoint 2 configure register**)**
CFG_EP3_1 [0x0816], CFG_EP3_0 [0x0817] (endpoint 3 configure register**)**
CFG_EP4_1 [0x0818], CFG_EP4_0 [0x0819] (endpoint 4 configure register**)**
CFG_EP5_1 [0x081a], CFG_EP5_0 [0x081b] (endpoint 5 configure register**)**
CFG_EP6_1 [0x081c], CFG_EP6_0 [0x081d] (endpoint 6 configure register**)**
CFG_EP7_1 [0x081e], CFG_EP7_0 [0x081f] (endpoint 7 configure register**)**

Note: endpoint 0 is the control port. It occupies 64 bytes xram space the size and mode of it cannot be configured.

Next is the detail description of the register：

EP_ADDR_MSB（the MSB address bit of endpoints）：

EP_ADDR_MSB	7	6	5	4	3	2	1	0

0x0840	-	-	-	-	-	-	-	-

Table 30 USB MSB endpoint address

7: the MSB of endpoint 7 address. It decides the port address locates above 1K space or below it. (must set to 0 for BK51XX)

6: the MSB of endpoint 6 address. It decides the port address locates above 1K space or below it. (must set to 0 for BK51XX)

5: the MSB of endpoint 5 address. It decides the port address locates above 1K space or below it. (must set to 0 for BK51XX)

4: the MSB of endpoint 4 address. It decides the port address locates above 1K space or below it. (must set to 0 for BK51XX)

3: the MSB of endpoint 3 address. It decides the port address locates above 1K space or below it. (must set to 0 for BK51XX)

2: the MSB of endpoint 2 address. It decides the port address locates above 1K space or below it. (must set to 0 for BK51XX)

1: the MSB of endpoint 1 address. It decides the port address locates above 1K space or below it. (must set to 0 for BK51XX)

0: the MSB of endpoint 0 address. It decides the port address locates above 1K space or below it. (must set to 0 for BK51XX)

Default 1, above 1K space.

CFG_EP0_1 (the configure register 1 of endpoint 0)：

CFG_EP0_1	7	6	5	4	3	2	1	0
0x0810	dir	ep0_en	/				addr[9:8]	

Table 31 Configure Register 1 of Endpoint 0

7. Dir, port direction

1: IN (BK5100 send out data)；

0: OUT (the PC send out data)。

6. ep0_en，endpoint 0 enable

When ep_rdy[0] =0 and ep0_en=0, usb no respond to external now.

1-0. addr[9:8] ： The higher 2 bits ([9:8]) address of endpoint0. The low 8 bits address is stored in CFG_EP0_0.

Note: The dir bit of CFG_EP0_1 is set or cleared by software except that it is cleared by hardware when SETUP token coming. The direction is forced to OUT to access 8 bytes setup request in this condition. The setup request has the highest priority.

CFG_EPn_1 (endpoint n configure register)：(n=1 - 7)

CFG_ EPn_1	7	6	5	4	3	2	1	0
	Dir	Mode		Size			Addr[9:8]	

Table 32 Endpoint n Configure Register

7. Dir:

1: IN
0: OUT

6-5. Mode:
0 -- Control Transfer
1 -- Bulk Transfer
2 -- ISO Transfer
3 -- Interrupt Transfer

4-2. Size：
0— endpoint not available
1— 16 bytes buffer size
2—32 bytes buffer size
3—64 bytes buffer size
4—128 bytes buffer size
5—256 bytes buffer size
6—512 bytes buffer size
7—endpoint not available

1-0. Addr[9:8] ： The higher 2 bits ([9:8]) address of endpoint n.

CFG_EPn_0 (configure register 0 of endpoint n)：(n=0 - 7)

CFG_ EPn_0	7	6	5	4	3	2	1	0
	addr[7:0]							

Table 33 Endpoint nConfigure Register 0

The lower 8 bits address of endpoint n.

14.4 Interrupt

External interrupt 4 is assigned to USB. Int4 will be triggered if any enabled interrupt bit in USBINT0 or USBINT1 is set to 1. Software should query the register to find out the relevant interrupt source. Also, software should clear the interrupt bit by set it to 1 after dealing with the interrupt.

USBINT0 interrupt register

USB INT0	7	6	5	4	3	2	1	0
0x080a	ctl_ rec	ctl_ send	rx_rdy	tx_rdy	usb_rst	usb_su s	usb_re s	usb_so f

Table 34 USBINT0 Interrupt Register

7. ctl_rec: data received on control port (endpoint 0)
6. ctl_send: data send on control port (endpoint 0)
5. rx_rdy： data received on endpoint 1-7
4. tx_rdy ： data send on endpoint 1-7
3. usb_reset ： USB Reset interrupt

2. usb_sus ： USB suspend interrupt
1. usb_res：USB resume interrupt。
0. usb_sof： USB Start Of Frame interrupt

When ctl_rec, rx_rdy or tx_rdy triggered，need to query EP_STATUS register for detail information.

EP_STATUS_IN (set by hardware and cleared by software)

EP_ STATUS	7	6	5	4	3	2	1	0
0x080e	EP7	EP6	EP5	EP4	EP3	EP2	EP1	sudat

Table 35 EP_STATUS Register

7. EP7：indicate tx_rdy is triggered by endpoint 7
6. EP6：indicate tx_rdy is triggered by endpoint 6
5. EP5：indicate tx_rdy is triggered by endpoint 5
4. EP4：indicate tx_rdy is triggered by endpoint 4
3. EP3：indicate tx_rdy is triggered by endpoint 3
2. EP2：indicate tx_rdy is triggered by endpoint 2
1. EP1：indicate tx_rdy is triggered by endpoint 1
0. Reserved

EP_STATUS_OUT (set by hardware and cleared by software)

EP_ STATUS	7	6	5	4	3	2	1	0
0x80F	EP7	EP6	EP5	EP4	EP3	EP2	EP1	sudat

Table 36 EP_STATUS Register

7. EP7：indicate rx_rdy is triggered by endpoint 7
6. EP6：indicate rx_rdy is triggered by endpoint 6
5. EP5：indicate rx_rdy is triggered by endpoint 5
4. EP4：indicate rx_rdy is triggered by endpoint 4
3. EP3：indicate rx_rdy is triggered by endpoint 3
2. EP2：indicate rx_rdy is triggered by endpoint 2
1. EP1：indicate rx_rdy is triggered by endpoint 1
0. Sudat: indicate that 8 bytes set up package arrived

USBINT1 interrupt register，set by hardware and cleared by software（write 1 to clear it）.

USB INT1	7	6	5	4	3	2	1	0
0x080b	bad_tok	crc16_er	overtim	pid_err	/	/	/	/

	en	r	e					

Table 37 USBINT1 Interrupt Register

7. bad_token: unsupported token received
6. crc16_err : the package received crc16 check error
5. overtime： timeout interrupt(no data received after OUT token or no ACK received after IN token)
4. pid_err ： endpoint1-7 transfer PID error interrupt

USB_EN0, USB_EN1 interrupt enable register (only can be read or write by software)

USB_ EN0	7	6	5	4	3	2	1	0
0x080c	ctl_rec_en	ctl_send_en	rx_rdy_en	tx_rdy_en	usb_rst_en	usb_sus_en	usb_res_en	usb_sof_en

Table 38 USB_EN0 Interrupt Enable Register

7. ctl_rec_en : data received on control endpoint 0 interrupt enable bit
6. ctl_send_en : data sent on control endpoint 0 interrupt enable bit
5. rx_rdy_en： data received on endpoint 1-7 interrupt enable bit
4. tx_rdy_en： data sent on endpoint 1-7 interrupt enable bit
3. usb_reset_en: USB Reset interrupt enable bit
2. usb_sus_en： USB suspend interrupt enable bit
1. usb_res_en： USB Resume interrupt enable bit
0. usb_sof_en： USB Start Of Frame interrupt enable bit

USB_ EN1	7	6	5	4	3	2	1	0
0x080d	bad_token_en	crc16_err_en	overtime_en	pid_err_en	/	/	/	/

Table 39 USB_EN1Interrupt Enable Register

7. bad_token_en: unsupported token received interrupt enable bit
6. crc16_err_en : the package received crc16 check error interrupt enable bit
5. overtime_en： timeout interrupt enable bit
4. pid_err_en ： endpoint1-7 transfer PID error interrupt enable bit

14.5 FIFO

It has been described that how to configure the register above. The next will depict how to use their register and how to operate them.

14.5.1 FIFO SFR register

(1) EP_RDY： (endpoint ready register)

EP_RDY	7	6	5	4	3	2	1	0
0x0821	ep7_rdy	ep6_rdy	ep5_rdy	ep4_rdy	ep3_rdy	ep2_rdy	ep1_rdy	ep0_rdy

Table 40 FIFO EP_RDY Register

Epn_rdy (n=1-7): endpoint n is ready for transferring USB data now.\
Cleared by hardware and set by software.

Note: Ep0_rdy is not same with Epn_rdy. It will be forced to 1 by hardware when setup token coming to receive 8 bytes setup request. (setup has the highest priority for USB protocol)

When Epn_rdy=0, device will send back NACK pakage for PC's IN/OUT request to indicate not ready now.

(2) FIFO capacity counters

If one endpoint has been configured as IN direction, software need write the length number into the FIFO capacity register to tell USB the package length need send.

When one port configured as OUT direction, software can read out the package length from the counter register once one package received successfully. (the unit is byte)

Every endpoint use 2 bytes register, so total 16 registers are occupied which are descripted as follow:

CNTn : the lower 8 bits FIFO counter register of endpoint n

CNTn	7	6	5	4	3	2	1	0
	counter [7:0]							

Table 41 FIFO lower 8 bits counter register

CNTn_HBIT: the upper 2 bits FIFO counter register of endpoint n

CNTn_HBIT	7	6	5	4	3	2	1	0
	/						counter [9:8]	

Table 42 FIFO upper 2 bits counter register

All the 16 registers address：

CNT0 [0x0823]
CNT0_HBIT [0x082b]
CNT1 [0x0824]
CNT1_HBIT [0x082c]
CNT2 [0x0825]
CNT2_HBIT [0x082d]
CNT3 [0x0826]
CNT3_HBIT [0x082e]

CNT4 [0x0827]
CNT4_HBIT [0x082f]
CNT5 [0x0828]
CNT5_HBIT [0x0830]
CNT6 [0x0829]
CNT6_HBIT [0x0831]
CNT7 [0x082a]
CNT7_HBIT [0x0832]

(3) EP_HALT(endpoint suspend register)

EP_ HALT	7	6	5	4	3	2	1	0
0x0820	ep7 _halt	ep6 _halt	ep5 _halt	ep4 _halt	ep3 _halt	ep2 _halt	ep1 _halt	ep0 _halt

Table 43 FIFO EP_HALT Register

epn_halt: the suspend flag of endpoint n. 1 indicates the endpoint has been suspended and this endpoint is not available now. This endpoint will send back STALL when IN/OUT token received to indicate it is not available now. (epn_halt can only be read/written by software except ep0_halt)

ep0_halt can be cleared by hardware. According to USB protocol, ep0_halt is cleared by hardware when setup token received to avoid that device can not receive control information.

14.5.2 FIFO Access

The access to FIFO is very simple for BK5100. Software can read or write the 2K EXRAM directly with MOVX instruction and without any register interface or control logic.

When using C language, you only need to initialize a start address for one endpoint FIFO which should be consistent with the address configured in endpoint regiser.

For example: the address of endpoint 1

addr[10:0]={**EP_ADDR_MSB.1, CFG_EP1_1[1:0], CFG_EP1_0** }

=

$$\mathbf{EP\_ADDR\_MSB.1}\times 2^{10}+\mathbf{CFG\_EP1\_1.1}\times 2^{9}+\mathbf{CFG\_EP1\_1.0}\times 2^{8}+\mathbf{CFG\_EP1\_0}$$

This 11 bits address can cover all the 2K EXRAM space from 0 to 0x7FF.

14.5.3 FIFO Operation

The above describe how to access the EXRAM by MCU, and, the USB part need access the EXRAM also. It will be explained next.

Accessing EXRAM by USB is implemented by DMA controller, and it is transparent to software.

According protocol, the host send out SETUP, IN and OUT token to request device

transfer. The device would start to transfer data after software inform device the relevant endpoint is “ready”. The DMA controller will write the received data into the FIFO assigned in the EXRAM (out endpoint), or read out the data that needed send to the host from FIFO (IN endpoint).

What time is ready? From software view, there are two cases:

1. The software had written the data needed send to host into FIFO. It is ready to send now.(IN)
2. The software had read out the data received from host from FIFO. It is ready to receive now.(OUT)

When it is ready, Software can set the corresponding EP_RDY to indicate it is ready now, and then USB will start to work automatically.

14.6 Device Address

The 7 bits function address is stored in FADDR register. The address is set by host through SET_ADDRESS command. The software should write the 7 bits address into FADDR after received this command. The address will act immediately after received SET_ADDRESS command. USB only can accept the data or token send to this address.

Device address(R/W by software only)

FUNCT_ ADDR	7	6	5	4	3	2	1	0
0x0822	/	function_addr [6:0]						

Table 44 device address register

14.7 Frame number register

FRAM_NO_0 : Frame number lower 8 bits （write by hardware, read by software only）

FRAM_ NO_0	7	6	5	4	3	2	1	0
0x0808	Frame number [7:0]							

Table 45 FRAM_NO_0 lower 8 bits register

FRAM_NO_1: Frame number upper 3 bits， （write by hardware, read by software only）

FRAM_ NO_1	7	6	5	4	3	2	1	0
0x0809	/					Frame number [10:8]		

Table 46 FRAM_NO_0 upper 3 bits register

14.8 USB power management

USB_PWR_CN ： USB power control register

USB_	7	6	5	4	3	2	1	0

PWR_CN								
0x0841	pu_en	DN	DP	/		usb_rst	usb_sus	remote_ wakeup

Table 47 USB power control register

Pu_en: PULL UP enable， D+（dp）pull up enable in chip. When it is disabled, device disconnect with outside circuit.

DN: indicate D+ logic level（can used to debug）.（read only）

DP : indicate D- logic level（can used to debug）.（read only）

Usb_rst: USB module will reset when write 1 into it. (Exclude control register)

USB_sus： USB module will enter low-power mode when write 1 into it. The USB protocol engineer is stopped and no response to outside. It is used as suspend state usually in USB protocol.（can R/W by software）

remote_wakeup：according USB protocol, the device with remote wakeup function can send wake up signal to host.（R/W）When it is set to 1, USB force D+ and D- into K state, and release it when clear it.

14.9 Endpoint Buffer

For a transfer without buffer, it is described as follow: (IN direction)

1. MCU write the first package into the endpoint buffer and set relevant EP_RDY.
2. Wait transfer command from host, and send out interrupt when transfer is done.
3. MCU responds to interrupt and enter into relevant interrupt application. Then write the next package into the buffer and set EP_RDY.
4. Wait transfer command from host, and recurrence as described before.

The USB bandwidth utilize efficiency is the main disadvantage for this transfer mode. The host should wait when mcu wrote data into FIFO, and MCU should wait when USB sent data out.

For this, multi-buffer mechanism is applied in BK5100. MCU can write next package into FIFO when the current pakage is sending. So, when transfer command coming, the data can be sent immediately.

For example, 2-buffer is implemented as follows:

Config EP1 as IN endpoint, the capacity of EP1 is 64byte.

1. Configure the start FIFO address of EP1 as 0x500 and depth is 0x40.
2. Write the first pakage into 0x0500-0x0540, and then set EP_RDY register to indicate the data is ready.
3. At once, write the next package into 0x0540-0x0580 and wait the send out interrupt coming.
4. When the first package transfer complete, configure the start address of EP1 as 0x540, and then set EP_RDY to indicate the data is ready.
5. When the send out interrupt come, back to step 1.

Like this, 3 4 5 …-buffer is also can be implemented.

15 RF part

A RF transceiver BK2425 is integrated in BK2450. All the feature and performance are identical with BK2425. Please refer to BK2425 datasheet for detail usage.

15.1 Electrical Specifications

RF PART

Name	Parameter (Condition)	Min	Typical	Max	Unit	Comment
	Operating Condition					
VDD	Voltage	1.9	3.0	3.6	V	
TEMP	Temperature	-40	+27	+85	ºC	
	Digital input Pin					
VIH	High level	0.7VDD		VDD+0.7	V	
VIL	Low level	VSS		0.3VDD	V	
	Digital output Pin					
VOH	High level (IOH=-0.25mA)	VDD- 0.3		VDD	V	
VOL	Low level(IOL=0.25mA)	0		0.3	V	
	Normal condition					
IVDD	Power Down current		3		uA	
IVDD	Standby-I current		50		uA	
IVDD	Standby-II current		300		uA	
	Normal RF condition					
FOP	Operating frequency	2400		2527	MHz	
FXTAL	Crystal frequency		16		MHz	
RFSK	Air data rate	250		2000	Kbps	
	Transmitter					
PRF	Output power		4		dBm	
PBW	Modulation 20 dB bandwidth(2Mbps)		TBD		MHz	
PBW	Modulation 20 dB bandwidth (1Mbps)		TBD		MHz	
PBW	Modulation 20 dB bandwidth (250Kbps)		TBD		KHz	
IVDD	Current at -25 dBm output power		9.8		mA	
IVDD	Current at -18 dBm output power		10.2		mA	
IVDD	Current at -12 dBm output power		10.8		mA	
IVDD	Current at −7 dBm output power		11.6		mA	
IVDD	Current at −1 dBm output power		13.4		mA	
IVDD	Current at 4 dBm output power		18		mA	
	Receiver					
IVDD	Current (2Mbps)		16.5		mA	
IVDD	Current (1Mbps)		16		mA	
IVDD	Current (250Kbps)		16		mA	
Max Input	1 E-3 BER		10		dBm	
RXSENS	1 E-3 BER sensitivity (2Mbps)		-88		dBm	
RXSENS	1 E-3 BER sensitivity (1Mbps)		-91		dBm	
RXSENS	1 E-3 BER sensitivity (250Kbps)		-96		dBm	

MCU part

Name	Parameter (Condition)	Min	Typical	Max	Unit	Comment
	Operating Condition					
VDD	Voltage	1.9	3.0	3.6	V	
TEMP	Temperature	-40	+27	+85	ºC	
	Digital input Pin					
VIH	High level	0.7VDD		5.25	V	
VIL	Low level	VSS		0.3VDD	V	
	Digital output Pin					
VOH	High level (IOH=-0.25mA)	VDD- 0.3		VDD	V	

VOL	Low level(IOL=0.25mA)	0		0.3	V	
	Work mode current（MCU PART）					
	Sleep mode (RCOSC 32k)		4		uA	
	Idle mode at 16M		4		mA	
	Idle mode at 8M				mA	
	Idle mode at 4M				mA	
	Idle mode at XOSC32k(16M running)				mA	
	Active mode (16M)				mA	
	Active mode (8M)				mA	
	Active mode (4M)					
	Peripheral					
	LBD (always on)				uA	
	USB		2		mA	
	GPIO					
	Drive ability		4	8	mA	

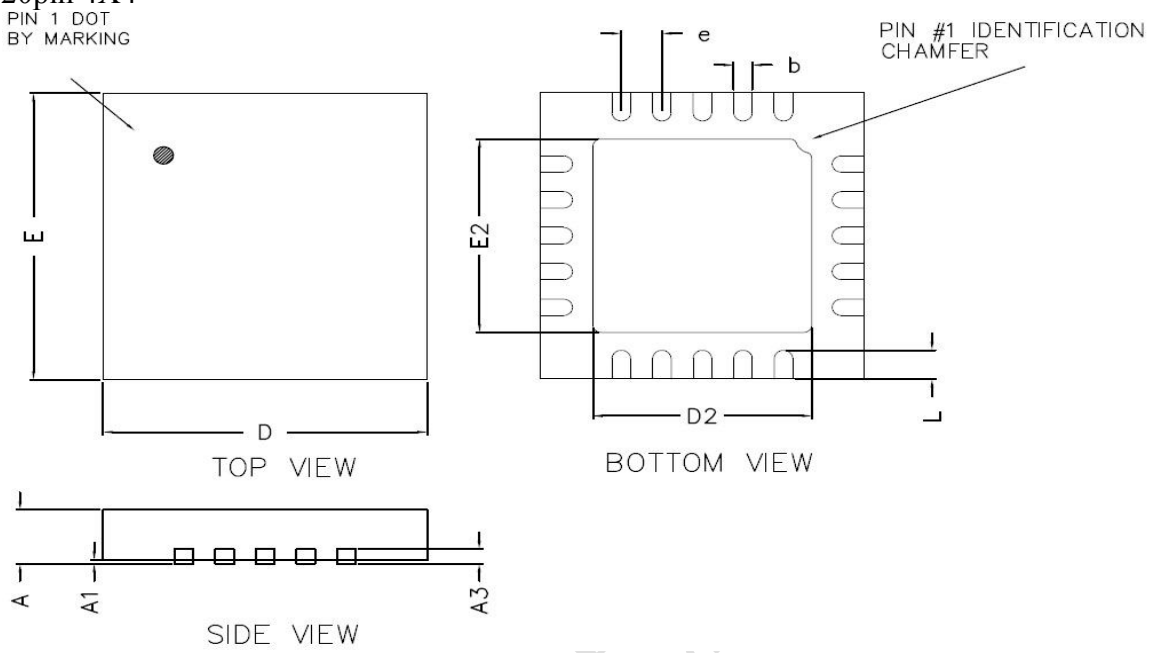
Table 48 MCU Electrical Specification

16 Typical Application Schematic

Please refer to the separate documents for detail.

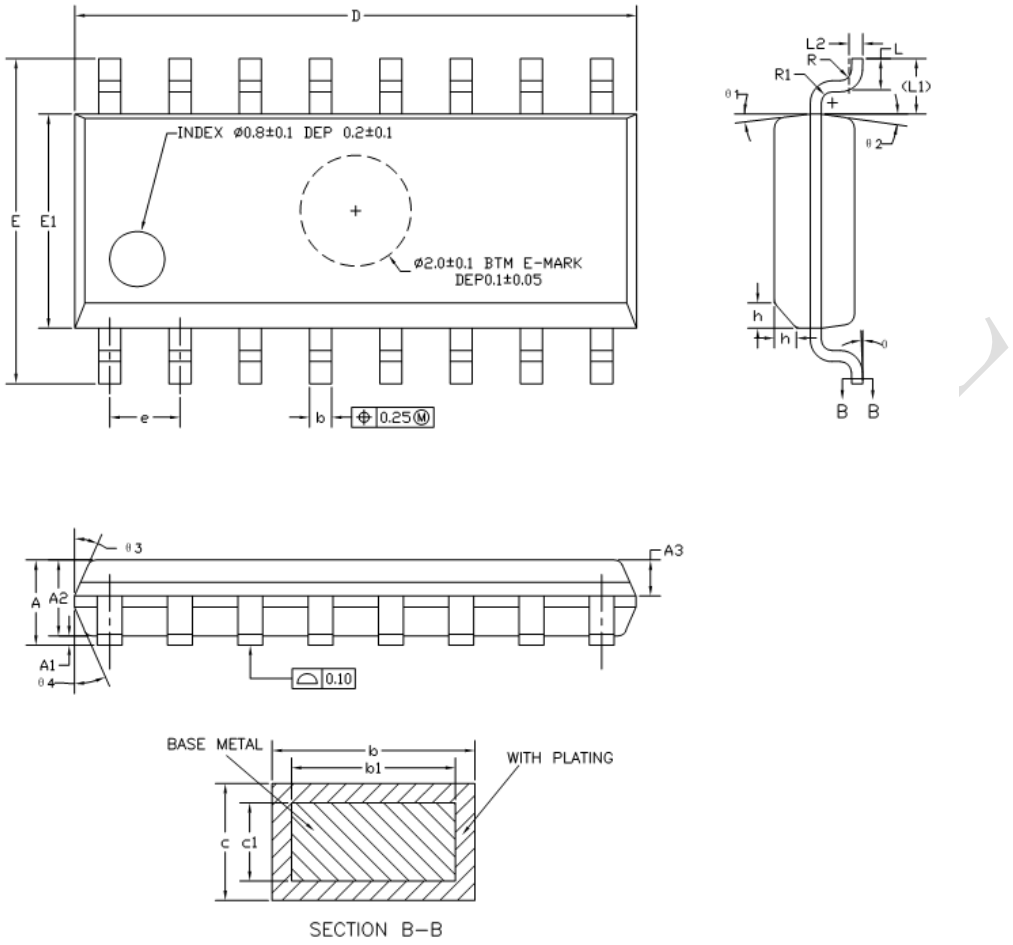
17 Package Information

20pin 4X4



Parameter	Min	Typ	Max	Unit
A	0.70	0.75	0.80	mm
A1	0.00	-	0.05	mm
A3		0.20 REF		mm
D	3.95	4.00	4.05	mm
E	3.95	4.00	4.05	mm
b	0.18	0.23	0.30	mm
L	0.30	0.40	0.50	mm
D2	2.55	2.70	2.80	mm
E2	2.55	2.70	2.80	mm
e		0.50 REF		mm

16pin SOP

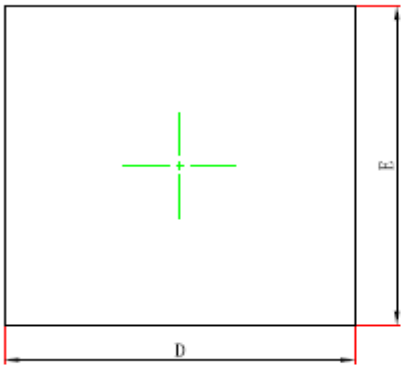


参数	最小	典型	最大	单位
A	1.35	1.60	1.75	mm
A1	0.10	0.15	0.25	mm
A2	1.25	1.45	1.65	mm
A3	0.55	0.65	0.75	mm
b	0.36	-	0.51	mm
b1	0.35	0.40	0.45	mm
c	0.17	-	0.25	mm
c1	0.17	0.20	0.23	mm
D	9.80	9.90	10.00	mm
E	5.80	6.00	6.20	mm
E1	3.80	3.90	4.00	mm
e	1.27 BSC			mm
L	0.45	0.60	0.80	mm
L1	1.04 REF			mm

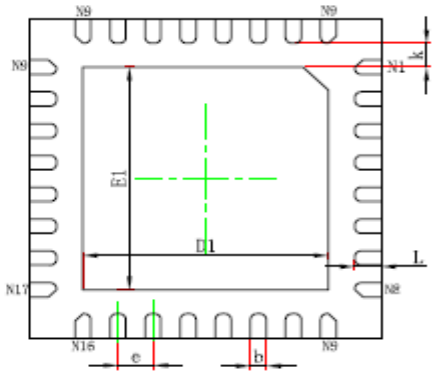
L2	0.25 BSC			mm
R	0.07	-	-	mm
R1	0.07	-	-	mm
h	0.30	0.40	0.50	mm
θ	0	-	8	°
θ1	6	8	10	°
θ2	6	8	10	°
θ3	5	7	9	°
θ4	5	7	9	°

QFN32-4X4

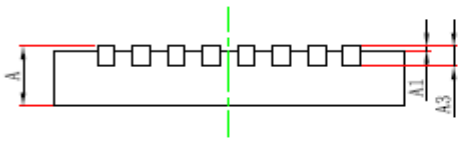
QFNWB4×4-32L-A (P0.40T0.75/0.85) PACKAGE OUTLINE DIMENSIONS



TOP VIEW



BOTTOM VIEW



SIDE VIEW

Symbol	Dimensions In Millimeters		Dimensions In Inches	
	Min.	Max.	Min.	Max.
A	0.700/0.800	0.800/0.900	0.028/0.031	0.031/0.035
A1	0.000	0.050	0.000	0.002
A3	0.203REF.		0.008REF.	
D	3.924	4.076	0.154	0.160
E	3.924	4.076	0.154	0.160
D1	2.700	2.900	0.106	0.114
E1	2.700	2.900	0.106	0.114
k	0.200MIN.		0.008MIN.	
b	0.150	0.250	0.006	0.010
e	0.400TYP.		0.016TYP.	
L	0.224	0.376	0.009	0.015

18 Solder Reflow Profile

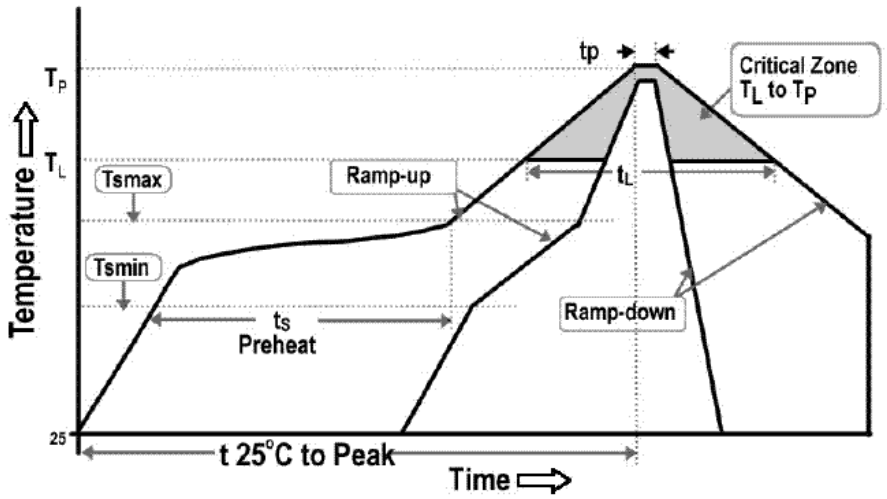


Figure 24 Classification Reflow Profile

Table 49 Solder Reflow Profile

Profile Feature		Specification
Average Ramp-Up Rate (tsmax to tp)		3°C/second max.
Pre_heat	Temperature Min (Tsmin)	150°C
	Temperature Max (Tsmax)	200°C
	Time (ts)	60-180 seconds
Time Maintained above	Temperature (TL)	217°C
	Time (tL)	60-150 seconds
Peak/Classification Temperature (Tp)		260°C
Time within 5°C of Actual Peak Temperature (tp)		20-40 seconds
Ramp-Down Rate 6		6°C/second max.
Time 25°C to Peak Temperature 8		8 minutes max.

19 Order Information

20 Solder Reflow Profile

21 Contact Information

Beken Corporation Technical Support Center

Shanghai office
Suite 3A,1278 Keyuan Road, Zhangjiang High-Tech Park,
Pudong New District, Shanghai, P.R. China
Phone: 86-21-51086811,60871276
Fax: 86-21-60871277
Postal Code: 201203
Email: info@bekencorp.com
Website: www.bekencorp.com

Shenzhen office
Room 718,Shenzhen High-Tech Industrial Estate,
Nanshan, Shenzhen, P.R. China
Phone: 86-755-2655 1063
Postal Code: 518057